AD-A038 317 VIRGINIA POLYTECHNIC INST AND STATE UNIV BLACKSBURG --ETC F/G 13/13
RELIABILITY STUDY OF SINGER. VOLUME II. USER'S MANUAL.(U)
JAN 77 A E SOMERS, S M HOLZER, J C BRADSHAW F29601-75-C-0050
UNCLASSIFIED AFWL-TR-76-192-VOL-2 NL

1 OF 4
AD
A038317

AFWL-TR-76-192, Vol. II

AFWL-TR-
76-192
Vol. II

Nu

RELIABILITY STUDY OF SINGER

Volume II
User's Manual

DEPARTMENT OF CIVIL ENGINEERING
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

January 1977

Final Report

Approved for public release; distribution unlimited.

This research was sponsored by the Defense Nuclear Agency under Subtask Y99QAXSC157, Work Unit 04, Work Unit Title, Strategic Structures Vulnerability/Hardening.

Prepared for
Director
DEFENSE NUCLEAR AGENCY
Washington, DC 20305

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base, NM 87117

1766

This final report was prepared by the Department of Civil Engineering, Virginia Polytechnic Institute, Blacksburg, Virginia, under Contract F29601-75-C-0050, Job Order 5710WDNS, with the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico. The former Project Officer was Lt Rodney Galloway. Lt Bradshaw (DES) was the Laboratory Project Officer-in-Charge.

This technical report has been reviewed and is approved for publication.

JOEL C. BRADSHAW III
Lieutenant, USAF
Project Officer

FOR THE COMMANDER

JAMES M. WARREN
Lt Colonel, USAF
Chief, Survivability Branch

FRANK J. LEECH
Lt Colonel, USAF
Chief, Civil Engineering Research Division

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFWL-TR-76-192, Vol. II

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): 6 RELIABILITY STUDY OF SINGER, Volume II, User's Manual,

5. TYPE OF REPORT & PERIOD COVERED: 9 Final Report.

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): 10 A. E. Somers, S. M. Holzer, J. C. Bradshaw & R. G. Galloway

8. CONTRACT OR GRANT NUMBER(s): 15 F29601-75-C-0050

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Department of Civil Engineering, Virginia Polytechnic Institute & State Univ., Blacksburg, Virginia 24061

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62704H, 5710WDNS

11. CONTROLLING OFFICE NAME AND ADDRESS: Director, Defense Nuclear Agency, Washington, D.C. 20305

12. REPORT DATE: 11 January 1977

13. NUMBER OF PAGES: 374

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Air Force Weapons Laboratory (DES), Kirtland Air Force Base, NM 87117 12 369p.

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

18 AFWL 19 TR-76-192-Vol-2

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This research was sponsored by the Defense Nuclear Agency under Subtask Y99QAXSC157, Work Unit 04, Work Unit Title, Strategic Structures Vulnerability/Hardening.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Reinforced Concrete Analysis
Dynamic Analysis
Nonlinear Structural Response
Airblast Effects
Finite Element Method
Energy Minimization
Structural Failure & Collapse

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The second volume of this report is principally developed for the user, as an aid in the use and modification of the SINGER code. The objective is to document the experience gained through the study of problem solutions and code details. In addition, this volume includes the description of new subroutines developed and a listing of the current version of the code.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

404191 1B

PREFACE

This is the second volume of a two-volume report on the work performed under the contract F29601-75-C-0050.

CONTENTS

SECTION I

INTRODUCTION

The original version of the SINGER program was developed under contract F29601-73-C-0022. Since the completion of that phase of the development, significant experience has been gained working with the program. A variety of problems has been studied, including those with steel elements and those with reinforced concrete elements. Much of the detailed understanding acquired has been documented for future reference, and code modifications have been introduced to correct errors or to improve computational accuracy. The objective of this volume is to document these areas of study for future reference. The organization of the documentation is in the following form:

1. Additional user assistance for the preparation of input data and for the interpretation of results;
2. Additional program details not available in previous documents;
3. New subroutine documentation;
4. Listing of the program code in its current form.

SECTION II

ADDITIONAL USER ASSISTANCE

The objective of this section is to describe alterations and clarifications in the preparation of problem data for input and in the explanation of some output details consistent with the current status of the program. The input and output features of the original SINGER code were both documented in the User's Guide [1]. A broader understanding of the program has provided the basis for expanding this documentation for the benefit of the user. Input data preparation is discussed in Addenda to the User's Guide and Error Controls. Some output characteristics are discussed in Minimization and Convergence Problems and Strain Discontinuity.

ADDENDA TO THE USER'S GUIDE

Although the basic structure of the original User's Guide is still valid, some points require correction or clarification. In addition, a few modifications need to be noted. These items are listed below; the page numbers shown refer to the original User's Guide [1].

1. Corrections:

   a. (p. 23), footnote no. 3: The parameter k, which defines the characteristic of the drop-elastic unloading curve for concrete, must be greater than zero or less than or equal to 1 to be an acceptable value.

   b. (p. 34), footnote no. 4: If the user inputs the number of bars and the bar size number, subroutine

BARS will calculate the area, bar size diameter, and perimeter.

c. (p. 41), footnote no. 1: If no initial conditions are specified and IANAL = 0, a static analysis is performed according to specified load increments. If no initial conditions are specified and IANAL = 1, a static analysis is skipped and a dynamic analysis is performed.

d. (p. 42), card 3/R, columns 1 - 5: Format I5 should be format I4, 1X.

e. (p. 43), footnote no. 3; paragraph 3: For vertical slopes, i.e., $t_{i+1} - t_i = 0$ (where t represents time), the forcing function value is set to F_i in subroutine TABL.

2. Clarifications:

a. (p. 18), footnote no. 6: The maximum relative error for determining the converged state of minimization (SERR) cannot be less than $\sqrt{EPS}$, where currently EPS = 1.E - 14; otherwise an error message is printed. If this value is input as zero, the program sets its value to $\sqrt{EPS}$.

b. (p. 22-23), Material Data Block: User input material functions must be defined by seven points, each with a corresponding stress and strain value. None of these strain values can be in descending order. The concrete material must have negative stress and strain values,

but the crushing strength should be a positive number.

The stress value at point 5 is used to indicate a default function for confined concrete for stress and strain modification due to confinement effects. If this stress value is zero, the stress and strain values at point 5 and the strain at point 6 are computed; if this stress is nonzero, a user input function is indicated and no additional computations are needed. (Although this computation affects the material function, it is not done in the input sequence until after the stirrup data are known.)

Both unconfined and confined concrete material functions should be input for a problem using reinforced concrete elements. If a single concrete material is desired for the element, both functions can be input with identical values.

The steel stress-strain values input by the user should be positive.

The strain corresponding to the peak stress of the user input function for concrete is the reference point for determining the unloading characteristics; unloading prior to reaching this reference strain value is a straight line parallel to the initial segment slope (path a b in figure 1); unloading after passing this strain value follows the drop-elastic path determined by the parameter k, where $0 < k \leq 1.0$

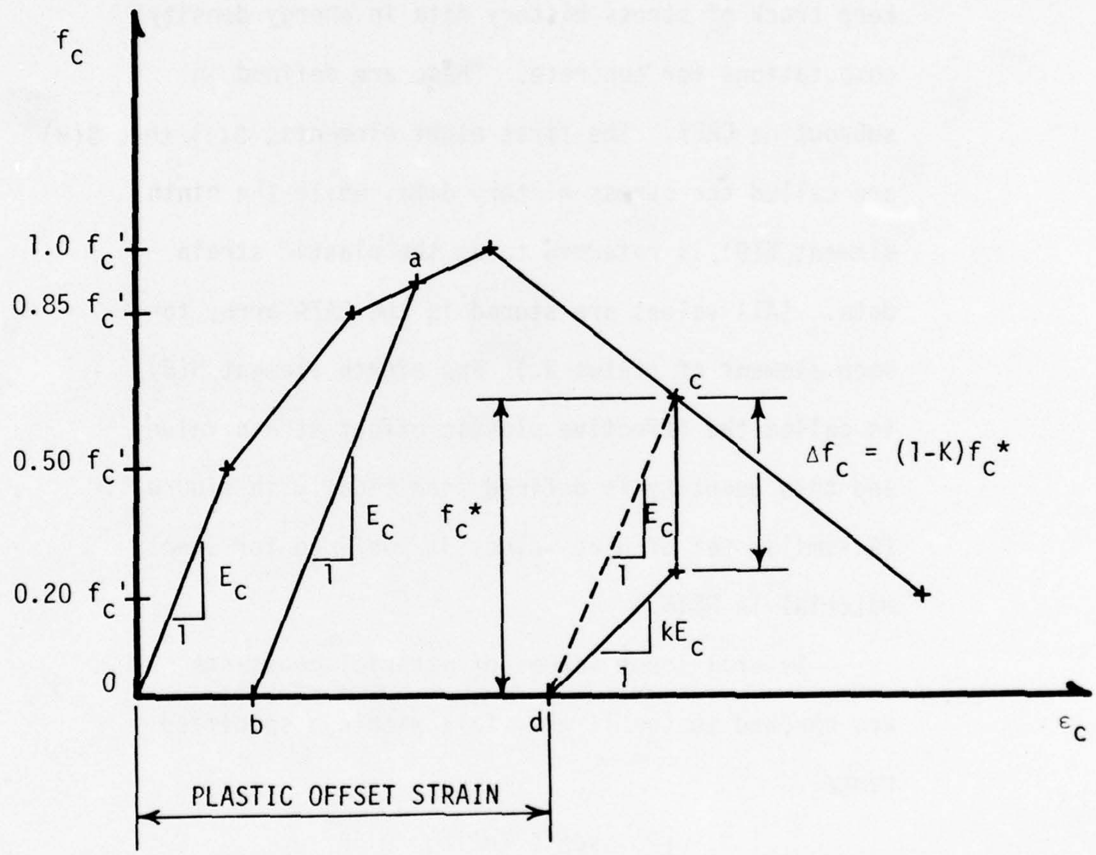


FIG. 1 CONCRETE MATERIAL PARAMETER DEFINITIONS

(path c d in figure 1). The drop-elastic parameter k is used to determine the length of the vertical stress segment as shown in figure 1.

The array S containing nine elements is used to keep track of stress history data in energy density computations for concrete. These are defined in subroutine CRET. The first eight elements, S(1) thru S(8) are called the stress history data, while the ninth element, S(9), is referred to as the plastic strain data. (All values are stored in the DATA array for each element of status 3.) The eighth element, S(8), is called the effective plastic offset strain value, and this quantity is defined specifically in figure 1. (A similar set of nine values is defined for steel material in REIN.)

Several input values of material constants are checked to see if they fall within a specified range.

$$0. \leqq [\text{Poisson's Ratio}] < 0.50$$

$$0. \leqq [\text{Shear modulus}] < 15.\text{x}10^6 \text{ psi}$$

$$0. \leqq [\text{Material density}] < 0.50 \text{ lb/in}^3$$

$$0. < [\text{Unloading constant}] \leqq 1.0$$

$$2.5\text{x}10^3\text{psi} \leqq \begin{bmatrix} \text{Unconfined concrete strength} \\ \text{Confined concrete strength} \end{bmatrix} \leqq 8.\text{x}10^3\text{psi}$$

$$33.\text{x}10^3\text{psi} \leqq [\text{Steel yield stress}] \leqq 75.\text{x}10^3\text{psi}$$

c. (p. 28-39), Element Data Block: *Each reinforced concrete* element requires the element parameter card and the concrete data card. It is advisable to have at least one longitudinal reinforcement group since no check is made to see if there is none, and some DO loops are executed over the number of groups. The lateral reinforcement card is optional. (Stirrups do not enter directly into the element energy computations.) An unreinforced section can be simulated by introducing a very small nonzero steel reinforcement group area.

Each steel wide flange element requires only the wide flange reinforcement card; and a leaf spring element (if used) requires only the leaf spring flexibility card.

The lateral reinforcement parameter A_v (p. 35) refers to the cross sectional area of the stirrup at a single location, i.e. the cross sectional area cut by a plane passing normal to the stirrup.

d. (p. 44), Forcing Function Data Block: To input a static load which is generated by a forcing function in prescribed load increments, the following data are required: joint no. (col. 1-5), direction indicator (col. 11), the reference function number (col. 16-19), k (col. 20), 1.0 (SF) (col. 21-30), and LA = 0, TA = 0, TP = 0 (col. 31-60). In a static analysis, time is used as a dummy variable for defining the domain of

the forcing function. The sign (+ or -) of the force should be associated with the scale factor value input in the Forcing Function Data Block (p. 44).

3. Modifications:

   a. (p. 17-18), Control Data Block: Card No. 2, Columns 21-25, have been changed from A1, 4X to 2A1, 3X. Column 22 will contain the item MPRINT = secondary print level flag.

      M = suppress paging

      S, d* = standard paging

   Card No. 3, Columns 6-10, have been changed from 5X to 3X, I2 with the addition of the variable IINITD, read initial guess of displacements (activated if this value is negative). This option may be selected if the user wishes to input an initial guess of the displaced configuration of the structure rather than have SINGER make the initial guess. The initial guess displacements should be placed at the end of the Forcing Function Data Block, after the zeros card, in 8E10.0 format. They should be in the order of the joint numbers and the order of X, Y, and rotation at each joint. The joint displacements should be followed by the internal node displacements for all the elements in the order of the element numbers. This gives a total of initial displacements equal to the number of degrees of freedom at the joints plus the number of elements. The displacements corresponding to the joints should be normalized by the

average element length. Columns 61-65 have been changed to I5 to add the variable INKS, output print option. This gives the user the choice to specify the time intervals at which the output will be printed. For example, if the user wants the results printed at every 10 time increments, the value 10 should be placed in columns 64-65. This will give results according to IPRINT, which is also specified at every 10 time increments. The intermediate time steps will display the time, function value, and displacements in a condensed nondimensional form.

b. The new subroutines DEFO and STRN, written to improve the directional properties of the element model (see section V, Volume I, and also section IV, Volume II), redefine the element deformation components. Since the existing subroutines LEAF and FAIL utilize the old component definitions, both should be studied for possible modifications before all four subroutines are used within the same program.

Also, these two new subroutines do not distinguish between finite and infinitesimal joint rotations. The same transformations are used regardless of the input parameter ILIN (p. 17).

Due to the function used to compute the arc tangent value in the new DEFO subroutine, values of joint rotations are valid within the range of ($-\pi<\theta<\pi$). Errors will be generated in this evaluation for rotations outside this range.

ERROR CONTROLS

In addition to inevitable roundoff errors, the dynamic response predictions of SINGER are subject to the following errors:

1. Temporal discretization error
2. Iteration error
3. Spatial discretization error
4. Quadrature error

The temporal discretization error is caused by the representation of the motion during a time step in terms of a finite power series in time, and the iteration error is caused by the termination of the search process after a prescribed accuracy has been achieved. These two errors are controlled automatically by SINGER [2].

The spatial discretization error is caused by the finite element representation of the continuum model of the beam-column, and the quadrature error is caused by the numerical integration of the internal-energy density over the volume of the finite element. These two errors must be controlled by the user to assure accurate energy predictions. A detailed description of error controls in energy evaluations is presented in section IV of Volume I of this report.

MINIMIZATION AND CONVERGENCE PROBLEMS

Listed below are the basic components of a general problem which may significantly influence the solution process:

1. Control parameters
2. Model characteristics
3. Load increments

1. Control parameters: The control parameters include the minimization convergence tolerance and the error measures. The state of convergence of the minimization process is controlled by the minimization tolerance parameter (SERR), a user input quantity which defines a limit to the stepsize used in a linear minimization. Also, an error value is computed which is intended to provide a measure of the quality of the converged state, referred to as the iteration error [2]. This error measurement is derived from the largest of the unbalanced generalized forces at the joints. In the program, this value is printed as the FULL TIME ERROR. For a static problem, this is the only error measure and it does not control the response in any way. For a dynamic problem, a half time error is computed, which is an error measure at the middle of the time step. It attempts to measure the truncation error due to approximate representation of the displacement function over a time step [2]. The size of the time step is controlled by the half time error.

The convergence tolerance parameter chosen too large can produce inaccurate results because the process terminates before a well defined minimum has been reached. A lower limit is placed on the parameter determined by the computer word size [1]. A range of values found to produce acceptable results for most problems is $1. \times 10^{-5}$ to $1. \times 10^{-7}$. It should also be noted that since the previous converged state is the basis for initialization of the next increment, there is some interaction between successive increments. However, the exact significance of this interaction with respect to solution accuracy is not well defined.

The error measures are computed during the solution process. It is possible to reach a converged state with respect to the convergence tolerance specified but still have a poor error measure. This may be due to a complex energy surface in the neighborhood of the minimum state chosen. Experience has shown that accurate results (measured by symmetry in the element response) are achieved when the full time error is smaller than $1. \times 10^{-2}$; a value of $1. \times 10^{-1}$ is questionable; and values larger than $1. \times 10^{-1}$ usually produce inaccurate response predictions. Good results are associated with the smaller values, such as $1. \times 10^{-4}$ or smaller.

The half time error measure in dynamic problems is compared with a tolerance $1. \times 10^{-3}$ (CRITU) for determining an acceptable converged state. (The magnitude of this error is usually larger than the full time error.) The time step is reduced to a magnitude of 0.60 of the previous value if this convergence limit is not achieved. The minimum time step defined in the code is $1. \times 10^{-7}$. It is also possible that the time step could be automatically increased to protect against roundoff error in computations. This occurs if the half time error is less than a fixed value, $1. \times 10^{-9}$ (CRITL). The next time interval is made 1.85 times the previous interval. (For a discussion of the dynamic error control logic, see Ref. [2], pages 50-53.)

2. Model characteristics: The most significant modeling characteristic which may influence the convergence process is the element distribution. The distribution chosen must accurately represent the continuum being modeled as reflected in how well the energy distribution can be reproduced. Extremely nonlinear material characteristics,

such as an elastic-plastic bilinear function, introduce complexities into the energy distribution, frequently occurring as localized energy concentrations at concentrated loads and totally restrained joints. Any region which is likely to develop large strain gradients should be subdivided to more accurately represent the energy concentration. It appears that members should be subdivided so that no one element contributes more than 50 percent to the energy function. The effect of poor element distribution can be reflected in the following ways: Large strain discontinuities may occur at the joints (see section VI, Volume I); and the accuracy of the continuum response prediction may be unsatisfactory.

3. Load increments: The magnitude of load increments can cause various convergence problems. A relatively large load increment may produce an equilibrium state so far removed from the previous state that convergence may be difficult to achieve, particularly if complexities due to nonlinear effects exist in the total energy surface. Also, the structure may reach a state of reduced stiffness so that a small change in load produces a relatively large change in the corresponding displacements. This has been observed in the steel wide flange beams and reinforced concrete beams loaded to failure (see section VI, Volume I), and also in the study of the elastica [3]. The effect of this difficulty can usually be reduced or eliminated by decreasing the load increments.

Another source of convergence difficulty occurs as the predicted response approaches the limit point of the equilibrium path. If the load is incremented to a value very near the limit point, an extremely

large displacement state may result with a poor error measure indicated. If the load level is incremented above the limit point, the solution search produces such extremely large values that the computer control system terminates the problem. There is no automatic termination check made in the program to detect this condition.

For most structures, the choice of load increments is not critical. The minimization algorithm appears to be a stable process and convergence is usually achieved for the load increments specified. The solution time may be longer for a problem containing a few large load steps as compared to one with several small load steps because the number of iterations for convergence for each large increment may be significantly increased. However, the solution accuracy may also be increased for some problems.

Two types of normal convergence may occur. The usual and most desirable type of convergence occurs when the stepsize used in a linear minimization is reduced to the specified tolerance. The other type occurs when a zero stepsize has occurred in the direction of a linear minimization. This type of convergence is indicated by the printed message NO MOTION IN THE LINEAR MINIMIZATION. This condition may occur during an apparently normal minimization process. It may also occur when an element yields. When yielding occurs, the minimization at that load level is repeated with the status of the yielded member changed. Since yielding may have occurred only at the outer fibers of the element, the strains at the integration points may still be within the elastic range. If this is the case, no change occurs in the solution vector and no motion is indicated.

Probably the most important consideration in obtaining accurate predictions of nonlinear structural response is the user's understanding of the system behavior. This is an indispensable aid in deciding how to subdivide the members and simplify the mass discretizations. It may be necessary to obtain satisfactory results through a trial and error procedure. Also, the user can study a test member to determine the subdivision necessary to accurately represent the expected load-deformation states (see section IV, Volume I).

STRAIN DISCONTINUITY

If one element is required to represent large changes in strain within its length (of the order of 50 times the yield strain or larger), a strain discontinuity may occur with respect to an adjacent element which is not so severely strained. Large strain gradients occur in regions of highly localized deformations, such as the region near a concentrated load laterally applied to a reinforced concrete beam after the steel reinforcement begins to yield. The discontinuities may appear not only as differences in magnitude, but also in sign. The solution process does not appear to be adversely affected by these discontinuities. However, since the stress resultants depend on the stress values computed across the sections at the ends of an element, the strain discontinuities produce stress resultant discontinuities.

The strain discontinuity problem can be improved by subdividing the member into smaller elements within the localized region. In this manner, each element is required to represent a smaller strain change along its length. Discontinuities may still occur, but their

magnitudes should be reduced since the energy distribution can be better represented. A more complete discussion with a demonstration of these effects is presented in section VI, Volume I, of this report.

SECTION III

ADDITIONAL PROGRAM DETAILS

Some program details have not been adequately explained in previous documentation. Such detailed descriptions are important for understanding the program structure, and they provide the basis for making additional modifications to the code. Program details discussed in this section include a description of selected variables, the organization of the array storage scheme called DATA and KDATA, and the procedure to be used for modifying the fixed storage allocation built into the program.

DESCRIPTIONS OF SELECTED VARIABLES

The variables described here are primarily those which determine the location of Gauss points and end points for both steel wide flange and reinforced concrete elements. The objective is to relate the names used in the code to the physical location of the points in the elements. The Gauss points define the locations at which energy density values are computed, and the end points are used in part as stress points from which to compute the element stress resultants. These point locations are shown in figure 6 for a reinforced concrete element and in figure 7 for a wide flange element (pages 31 and 34, respectively).

The location of a point for any element type is defined by an x coordinate and a y coordinate. The following list shows the variable names associated with the locations for both element types.

	Variable Names		
	X Location	Y Location	
		Gauss Point	End Point
Steel wide flange	XPI	YGP	YFIBR
Reinforcing bar	XPI	YBAR	YBAR
Concrete (confined and unconfined)	XPI	YGP	YFIBR

All elements use the same x location variable XPI as an array with the following form: XPI(J,M)   J = section number
M = element number

The section number refers to the cross sections located at the three Gauss coordinates in the longitudinal direction (for the Gauss point locations), and at the ends of the element for the end point locations.

The y location names are made identical where possible. The Gauss point locations for each type are named YGP, used as an array with the following form: YGP(I,M)   I = y location number
M = element number

The locations on a cross section are shown in figures 2 and 3 for the two element types. Points defined by I = 1, 2 and 6, 7 are at the Gauss locations for the flange or cover thickness. The remaining three points defined by I = 3, 4, and 5 are at the Gauss locations for the web or confined concrete depth.

The y locations for the end points use the same array name YFIBR with the following form: YFIBR(I,M)   I = y location number
M = element number

The locations are also shown on the cross sections of figures 2 and 3. There is a total of 11 points for each element type. The points

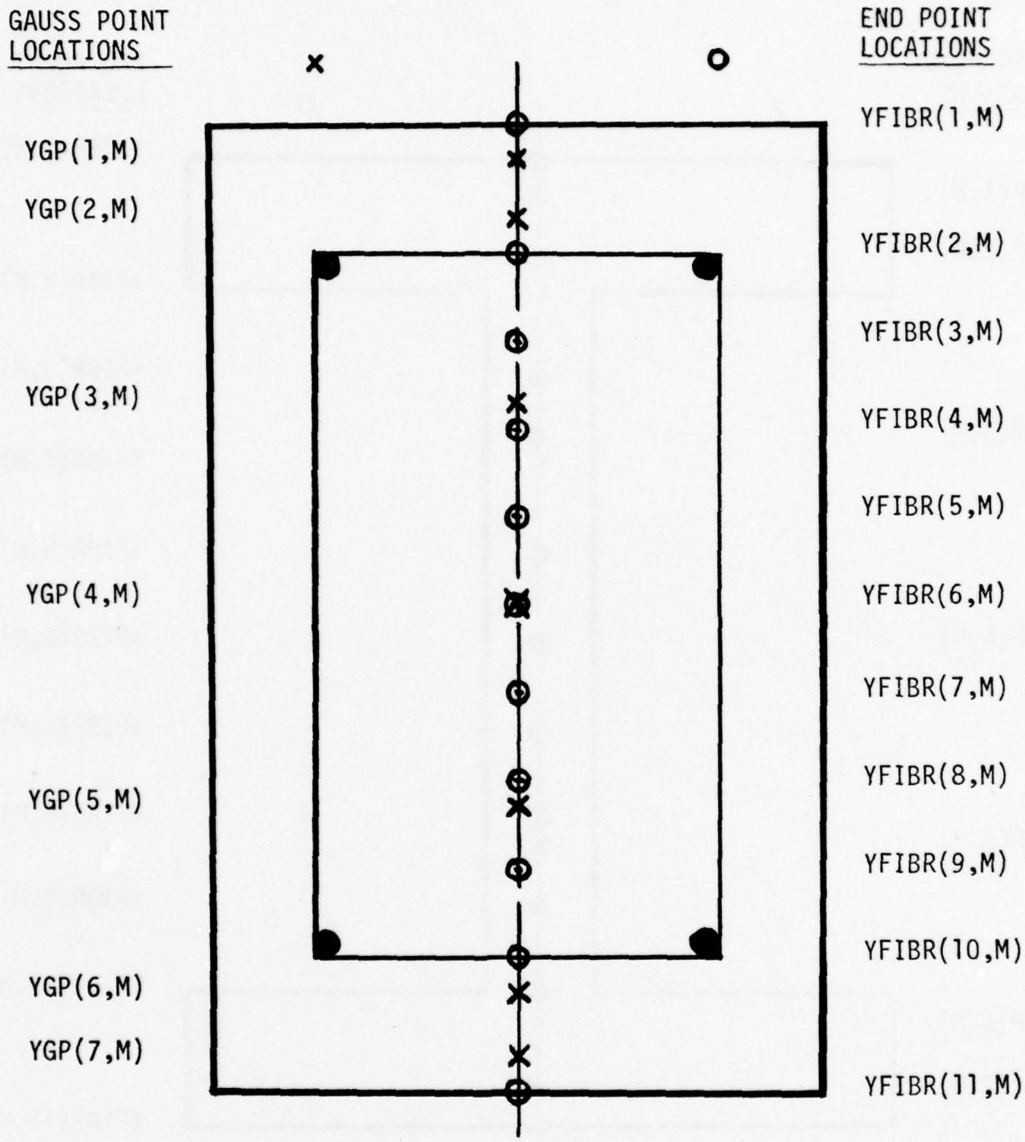


FIG. 2 GAUSS POINT AND END POINT LOCATIONS FOR A REINFORCED CONCRETE SECTION.

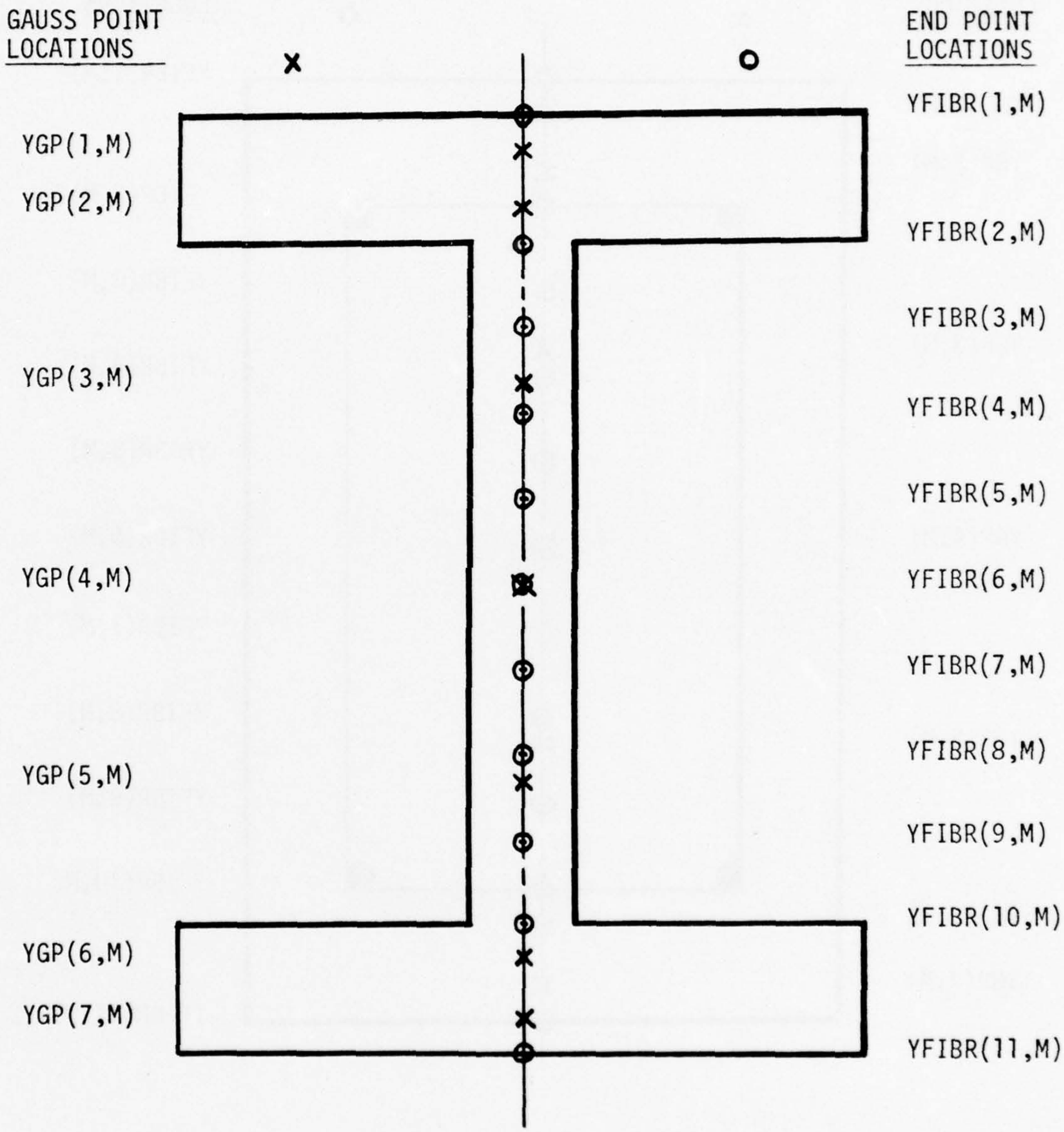


FIG. 3 GAUSS POINT AND END POINT LOCATIONS FOR A WIDE FLANGE SECTION.

defined by I = 1, 2 and 10, 11 are the limits of the flange or cover thickness. The points defined by I = 3, 4, 5, 6, 7, 8, and 9 are determined by dividing the web or confined concrete depth by 8 and spacing the points evenly at the eighth points.

The y locations for the longitudinal reinforcing bar groups use the array name YBAR with the following form:

YBAR(I,M) I = bar group number
M = element number

The YBAR values are the y locations of each longitudinal reinforcing bar group input for an element. Figure 6a shows the points on a reinforcing bar with the length the same as the element length. This form is valid for each bar group, regardless of the input length, since each one is redefined as an equivalent group over the element length for energy computations.

The determination of whether or not a reinforced concrete element has a confined core is based on the value stored in the array NTIES(M), where M is the element number. NTIES stores the number of tie (stirrup) groups input for a reinforced concrete element. If NTIES = 0, no ties exist for the element and there is no confined core. The total section is then treated as unconfined for energy computations. If NTIES is a positive integer, then there are ties within the element and the energy is computed for both the confined core and the unconfined outer shell.

ORGANIZATION OF DATA/KDATA STORAGE ARRAYS

Selected data values are stored in locations organized as two related arrays; one stores certain floating point data (DATA), while the other stores related integer data (KDATA). The DATA array includes the following values:

1. Curvature data
2. Reference forcing function data
3. Joint forcing function parameters
4. Plastic strain data
5. Stress history data

The KDATA array includes the following integer values:

1. Reference forcing function time points
2. Joint forcing function parameters
3. Plastic strain data location indices
4. Stress history location indices

The following discussion provides the organizational details for these two arrays.

1. Organization of the DATA Array: The DATA array contains floating point data which are necessary for the solution process. It contains curvature data that are used in subroutine SEEK; it includes function tables and forcing function data; this array also contains element stress-strain histories if an element has yielded (MSTAT = 3), or the user has input the option for storing the stress-strain historie (MSTAT = 3) rather than using the changeable status (MSTAT = 2). General storage in the DATA array is illustrated in figure 4.

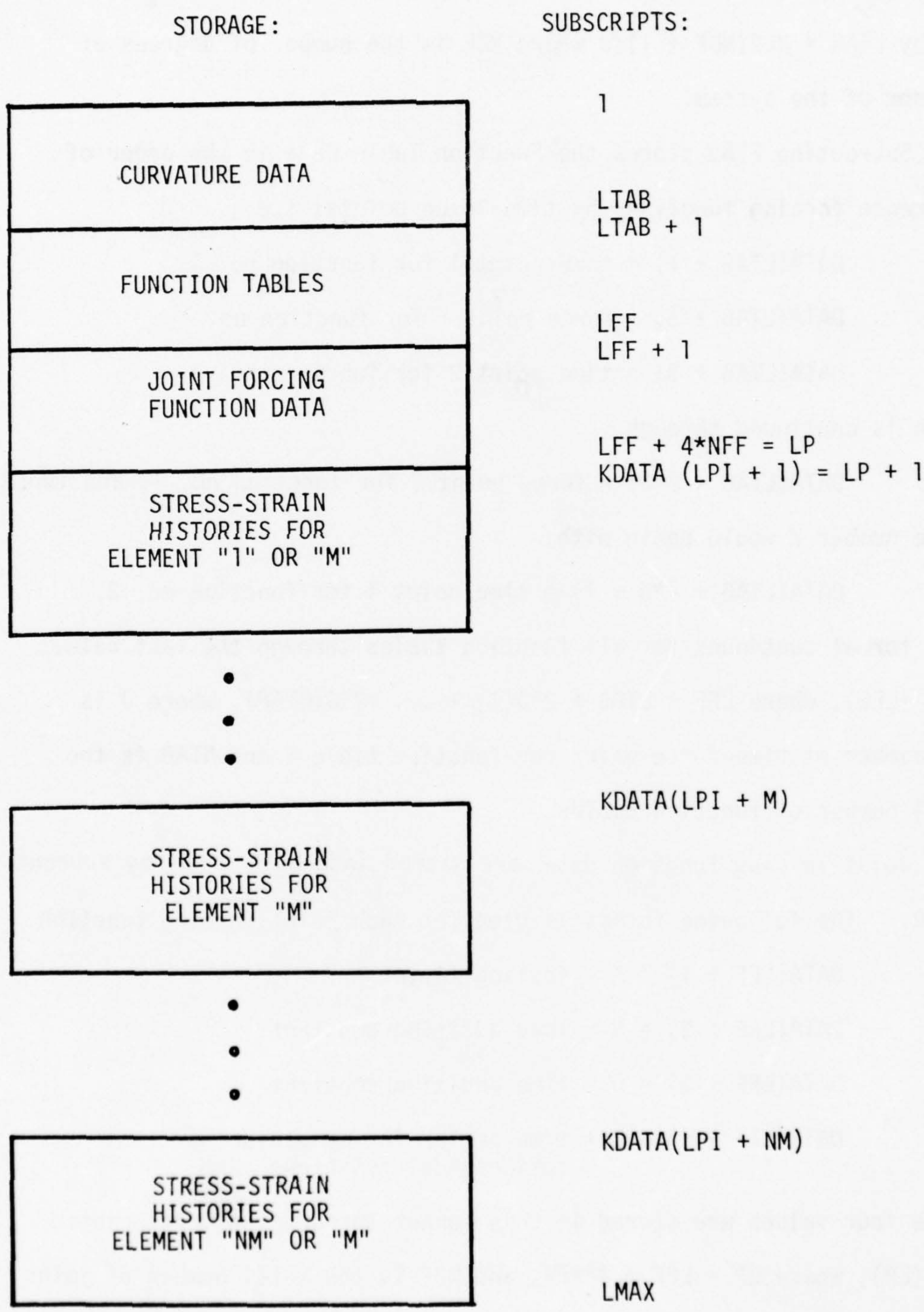


FIG. 4 GENERAL STORAGE IN THE DATA ARRAY.

Allocation of space in the first portion of the DATA array for curvature data is made in subroutine FTAB. The total allocation is set by LTAB = NDF(NDF + 1)/2 where NDF is the number of degrees of freedom of the system.

Subroutine FTAB stores the Function Table Data in the order of reference forcing functions by time-force points; i.e.,

DATA(LTAB + 1) = time point 1 for function no. 1

DATA(LTAB + 2) = force point 1 for function no. 1

DATA(LTAB + 3) = time point 2 for function no. 1

which is continued through

DATA(LTAB + 2*J) = force point J for function no. 1, and function table number 2 would begin with:

DATA(LTAB + 2*N + 1) = time point 1 for function no. 2.

This format continues for all function tables through the last value: DATA (LFF), where LFF = LTAB + 2*J(1) + ... +2*J(NTAB), where J is the number of time-force pairs for function table i and NTAB is the total number of function tables.

Joint forcing function data are stored into DATA array by subroutine JFOR. The following format is used for each joint forcing function:

DATA(LFF + 1) = A = scaling factor

DATA(LFF + 2) = B = load addition constant

DATA(LFF + 3) = C = time addition constant

DATA(LFF + 4) = D = time period for sinusoidal or cosinusoidal functions.

These four values are stored in this manner through the last point: DATA(LP), where LP = LFF + 4*NFF, and NFF is the total number of joint forcing functions input by the Forcing Function Data Block.

Element stress-strain data are stored in the array only if an element has yielded or the user has forced storage during input by setting IACT to N on the element parameter card in the Element Data Block. In most problems, not all elements will have yielded before collapse takes place. For this reason, only the elements requiring storage of their stress-strain data need the storage allocation, and these data are stored in the DATA array as each element yields. (The entry points for the stress and strain data are stored in the KDATA array.) A simple five element example where elements have yielded in the order of 3, 2, and 5 is shown in Figure 5. Reinforced concrete stress-strain data are stored in the following general form for each member:

REINFORCING STEEL STRAINS: 5*NGRP(M) points

UNCONFINED CONCRETE STRAINS: 31 points

CONFINED CONCRETE STRAINS: 27 points

REINFORCING STRESS HISTORIES: 40*NGRP(M) points

UNCONFINED CONCRETE STRESS HISTORIES: 248 points

CONFINED CONCRETE STRESS HISTORIES: 216 points

Temporary storage is made from the DATA array in the S array in subroutines COEN, STEN, and OUTS. Strain and stress data are updated through the S array, which is contained in the common block FIBER, in subroutines CRET and REIN. The S array contains nine values which reference one point within the element. The values S(1) through S(8) correspond to the stress history at a particular point, and S(9) contains the strain at that point within the element. These values are directly transferred between the S and DATA arrays when MSTAT is set

KDATA

SUBSCRIPT	STORAGE
LPI + 1	
LPI + 2	$LP_3 + 1$
LPI + 3	LP + 1
LPI + 4	
LPI + 5	$LP_2 + 1$
LPSI + 1	
LPSI + 2	$LP_3 + 1 + R_2$
LPSI + 3	$LP + 1 + R_3$
LPSI + 4	
LPSI + 5	$LP_2 + 1 + R_5$

DATA

ELEMENT	STORAGE	SUBSCRIPT
3	R_3 S_3	LP + 1 $LP + 1 + R_3$
2	R_2 S_2	$LP_3 + 1$ $LP_s + 1 + R_2$
5	R_5 S_5	$LP_2 + 1$ $LP_2 + 1 + R_3$

R_m = STRAIN DATA

S_m = STRESS DATA

LP_m = LAST POINT STORED

m = ELEMENT "m"

NOTE: ELEMENTS HAVE YIELDED IN THE ORDER 3, 2, 5; ELEMENTS 1 & 4 HAVE NOT YIELDED.

FIG. 5 RELATION BETWEEN DATA AND KDATA; 5-ELEMENT EXAMPLE.

to 3 for the element. If MSTAT is not set to 3 for the element, there is no transfer between these arrays at any of the points within the element.

Storage in the DATA array is made in the order of Gauss points (integration points) and force points (end points) for each of the six general groups of strain and stress listed above for each element. These point locations are defined in the preceding subsection, DESCRIPTIONS OF SELECTED VARIABLES.

Storage of the strain points of element M is made in the following way:

LP = KDATA(LPI + M) - 1 is the reference point as shown in Figures 4 and 5.

An example for two steel reinforcing bars is illustrated in Figure 6a. The storage for an element containing two bars is made as follows:

DATA (LP + 1)

LP + 2

LP + 3

LP + 4

LP + 5

LP + 6

where 1 through 6 refer to the Gauss points (x) in Figure 6a.

Storage continues with:

DATA (LP + 7)

LP + 8

LP + 9

LP + 10

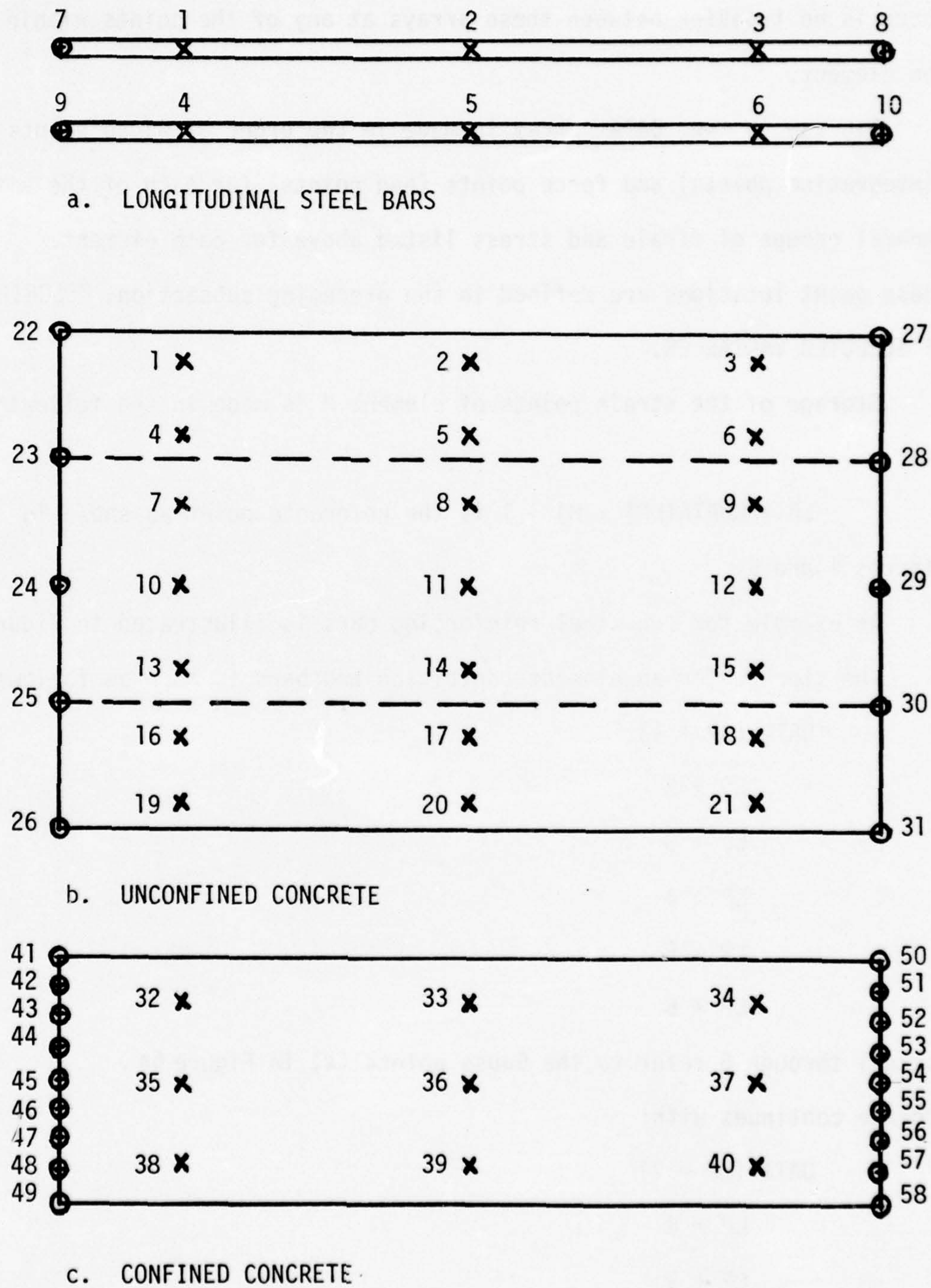


FIG. 6 STRESS-STRAIN STORAGE FOR REINFORCED CONCRETE ELEMENTS.

where 7 through 10 refer to the force points (o) in Figure 6a.

The concrete strain points are referenced by KRAIN = KDATA(LPI + M) + 5*NGRP(M) - 1, where NGRP(M) is the number of groups of reinforcing bars in element M. The subscripts for the unconfined and confined concrete in the DATA array continue from DATA(KRAIN + 1) to DATA(KRAIN + 58) as illustrated in Figures 6b and 6c.

The reinforcing steel stress histories are referenced by KRESS = KDATA(LPSI + M) - 1 which is illustrated in Figure 5. Eight values are stored for each of the Gauss points and force points in Figure 6a. In this example with two bars, the subscripts would continue with KRESS + 2 through KRESS + 48 for the Gauss points (1-6) and KRESS + 49 through KRESS + 60 for the force points (7-10). For each reinforcing bar group there are 24 storage locations for Gauss points and 16 locations for end force points which totals 40 points for each group.

The concrete stress histories are referenced by KRESS = KRESS + 40*NGRP(M). Eight values are stored at each of the points shown in Figures 6b and 6c. Storage is made for the unconfined concrete at the Gauss points with subscripts KRESS + 1 through KRESS + 168 which are 21*8 values. The force points are stored with subscripts KRESS + 169 through KRESS + 248 which are 5*8 + 5*8 values. Stress histories at the Gauss points of the confined concrete begin with the subscript KRESS + 249 and continue through KRESS + 320 which are 9*8 values. The force point stress histories continue from KRESS + 321 through KRESS + 464 which are 9*8 + 9*8 values.

Following is a summary of the storage in the DATA array of a reinforced concrete element, M, where NG = NGRP(M).

Strains	Values Stored	Total
Reinforcing steels:		
Gauss points	3*NG	3*NG
Force points	2*NG	5*NG
Unconfined concrete:		
Gauss points	21	5*NG + 21
Force points	10	5*NG + 31
Confined concrete:		
Gauss points	9	5*NG + 40
Force points	18	5*NG + 58

Stress Histories	Values Stored	Total
Reinforcing steels:		
Gauss points	24*NG	24*NG
Force points	16*NG	40*NG
Unconfined concrete:		
Gauss points	168	40*NG + 168
Force points	80	40*NG + 248
Confined concrete:		
Gauss points	72	40*NG + 320
Force points	144	40*NG + 464

Storage in the DATA array for a steel wide flange element is done in a similar way as that for a reinforced concrete element. The indexing for the strain data begins with KRAIN = KDATA(LPI + M) - 1. The Gauss point (x) subscripts range from KRAIN + 1 through KRAIN + 21 as shown in Figure 7. The subscripts for the force point strains range from KRAIN + 22 through KRAIN + 43. The stress histories are referenced

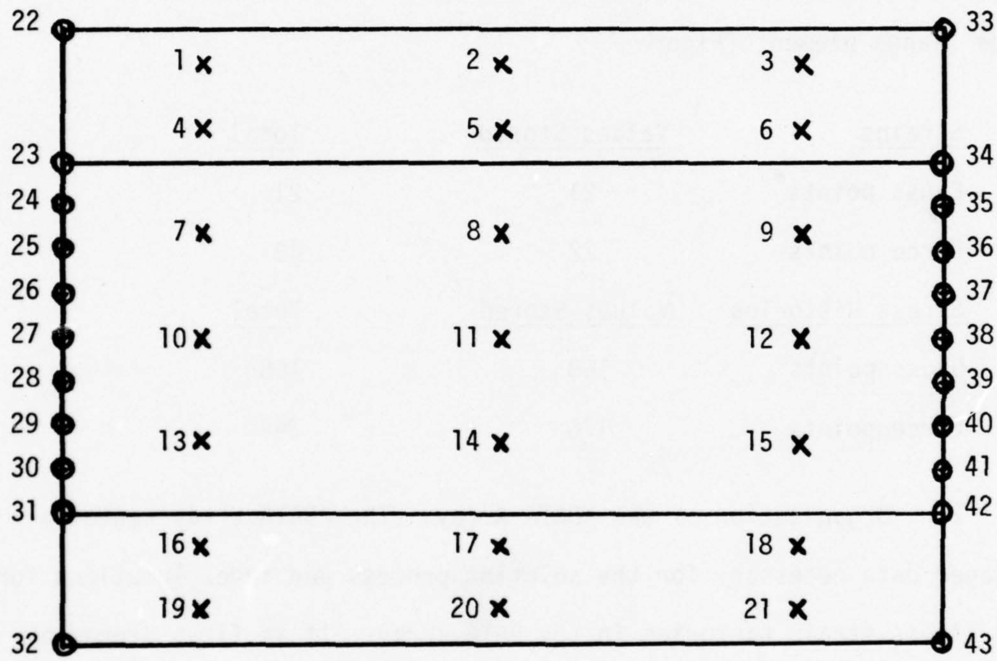


FIG. 7 STRESS-STRAIN STORAGE FOR WIDE FLANGE ELEMENTS.

by KRESS = KRAIN + 43. The subscripts for the Gauss point stress histories range from KRESS + 1 through KRESS + 168 which are 21*8 values. The stress history subscripts for the force points range from KRESS + 169 through KRESS + 344 which are 11*8 + 11*8 values.

The following is a summary of storage in the DATA array of a wide flange element (Figure 7):

Strains	Values Stored	Total
Gauss points	21	21
Force points	22	43
Stress Histories	Values Stored	Total
Gauss points	168	168
Force points	176	344

2. Organization of the KDATA Array: The KDATA array contains integer data necessary for the solution process and index locations for the stress-strain histories in the DATA array. It is first stored by subroutine FTAB with function table data. The variable LTABI, which is the load table index, is the locator for each function table input into the program through the Function Table Data Block. This variable is initialized to zero in ABET. The function table data are stored in the following format:

KDATA(LTABI + 1) = LTABI + 1

KDATA(LTABI + 2) = number of time points for each reference forcing function through the last forcing function table: KDATA(LFFI),

where LFFI = LTABI + 2*NTAB

and NTAB is the total number of reference forcing functions.

Forcing function data are stored in KDATA in subroutine JFOR through input in the Forcing Function Data Block. Storage is in the following format:

KDATA(LFFI + 1) = loaded joint number

KDATA(LFFI + 2) = reference joint number

KDATA(LFFI + 3) = direction

KDATA(LFFI + 4) = reference function number

KDATA(LFFI + 5) = last forward time point used for each joint forcing function through the last input forcing function: KDATA(LPI),

where LPI = LFFI + 5*NFF and NFF is the total number of joint forcing functions used.

The index or subscript values that define the locations of the element stress-strain histories in the DATA array are set in subroutine STOR. The following format is used:

KDATA(LPI + 1) = start of strain data for element 1

KDATA(LPI + 2) = start of strain data for element 2, and for each element, where KDATA(LPSI) is the starting point for the strain data of element NM and LPSI = LPI + NM.

KDATA(LPSI + 1) = start of stress history data for element 1

KDATA(LMAXI) = start of stress history data for element NM

where LMAXI = LPSI + NM = LPI + 2*NM.

MODIFICATION OF FIXED STORAGE ALLOCATION

When a problem is sufficiently large or computer storage is critical, various arrays within SINGER must be modified to accommodate these restrictions. The key to successful storage modification is an understanding of the several key variables which describe in general the

problem size in question. These variables are defined as follows:

Variable	Definition	Current Value
NDFD	Maximum number of degrees of freedom	90
NJD	Maximum number of joints	50
NLD	Maximum number of leaf springs	20
NMATD	Maximum number of materials	9
NMAX	Maximum index of DATA array	10000
NMAXI	Maximum index of KDATA array	500
NMD	Maximum number of members	45

Every change of storage within SINGER can be accomplished by changing those arrays which depend on these seven variables. Other program variations are possible, but a complete reprogramming would be necessary. These include having greater than 10 groups of reinforcing steel in one member, varying the Gaussian quadrature system in the program, and varying the number of elemental distortion components (varying the number of internal nodes).

Table 1 gives a list of the common storage data arrays and the variable arrays that must be changed. Table 2 gives other variables that must be dimensioned differently. Table 3 shows where the different common blocks are found within the subroutines. Table 4 gives example sizes for linear and nonlinear problems of the KDATA and DATA arrays.

Given a fixed storage requirement, the largest problem that can be solved by SINGER would consist of one where the material response is strictly linear in nature. All arrays, even though they are not used by a particular problem, should have a minimum value of 1.

TABLE 1 - COMMON VARIABLE ARRAYS

LOCATION	VARIABLE ARRAY	CURRENT VALUE	NECESSARY VALUE
COMMON	DATA	10000	NMAX
	KDATA	500	NMAXI
COMMON/CONBK	COAREA	4,45	4,NMD
	SIGMA	5,45	5,NMD
COMMON/ELEMENT	IP	45	NMD
	IPL	45	NLD
	IQ	45	NMD
	IQL	45	NLD
	MATR	45	NMD
	MATW	45	NMD
	MBAR	10,45	10,NMD
	MCODE	45	NMD
	MSHEAR	45	NMD
	MSTAT	45	NMD
	MTIES	45	NMD
	MTYPE	45	NMD
	NGRP	45	NMD
	NSPACE	6,45	NMD
	NTIES	45	NMD
COMMON/FIBER	DENS	9	NMATD
	EC	9	NMATD
	EPSU	9	NMATD
	ET	9	NMATD
	FCFY	9	NMATD

	G	9	NMATD
	PR	9	NMATD
	SLOPG	8,9	8,NMATD
	STN	8,9	8,NMATD
	STS	8,9	8,NMATD
	UNLK	9	NMATD
	ICODE	9	NMATD
	NAME	9	NMATD
COMMON/JOINTS	ACC	3,50	3,NJD
	BET	3,50	3,NJD
	DAS	3,50	3,NJD
	DIS	3,50	3,NJD
	ERJF	3,50	3,NJD
	ERJH	3,50	3,NJD
	ERJZ	3,50	3,NJD
	F	3,50	3,NJD
	FOR	3,50	3,NJD
	VEL	3,50	3,NJD
	X	50	NJD
	XDJ	3,50	3,NJD
	Y	50	NJD
	DER	3,50	3,NJD
	RESENG	3,50	3,NJD
	IDFI	90	NDFD
	IDFII	90	NDFD
COMMON/MEMBER	AGRP	10,45	10,NMD
	ATIES	6,45	6,NMD

BMEM	45	NMD
BPP	45	NMD
BDM	10,45	10,NMD
BWF	45	NMD
D	45	NMD
DP	45	NMD
DPP	45	NMD
DWF	45	NMD
EFFL	10,45	10,NMD
EFLM	45	NMD
HMEM	45	NMD
HTOP	45	NMD
HTWF	45	NMD
PDP	7,45	7,NMD
SPRING	5,20	5,NLD
STIES	7,45	7,NMD
TFWF	45	NMD
TWWF	45	NMD
UDM	45	NMD
URM	45	NMD
XBEG	10,45	10,NMD
XBEGM	45	NMD
XBEGS	6,45	6,NMD
XL	45	NMD
XPI	5,45	5,NMD
YBAR	10,45	10,NMD
YGP	7,45	7,NMD

	YFIBR	11,45	11,NMD
	YLDS	45	NMD
	XDM	45	NMD
	PDF	7,45	7,NMD
	DISM	45	NMD
COMMON/SAVEBK	SAVACC	3,50	3,NJD
	SAVAXL	2,45	2,NMD
	SAVCRV	2,45	2,NMD
	SAVMOM	2,45	2,NMD
	SAVSHR	2,45	2,NMD
	SAVSRP	3,20	3,NLD
	SAVSRQ	3,20	3,NLD
	SAVXDJ	3,50	3,NJD
	SAVVEL	3,50	3,NJD
	SVSTRN	12,45	12,NMD
	SVSTRS	12,45	12,NMD
COMMON/SEEKBK	DEFOR	90	NDFD
	STPSIZ	90	NDFD
	GRAD	90	NDFD
	GRADI	90	NDFD
	DELTAG	90	NDFD
	DIRECT	90	NDFD
	DIAG	90	NDFD

TABLE 2 - OTHER VARIABLES

SUBROUTINE	VARIABLE (CURRENT SIZE)	NECESSARY VALUE
ABET	SOLN(90)	NDFD
	NDFD=90	NDFD
	NJD=50	NJD
	NLD=20	NLD
	NMATD=9	NMATD
	NMAX=10000	NMAX
	NMAXI=500	NMAXI
	NMD=45	NMD
ADYN	SOLN(90)	NDFD
ASAN	SOLN(90)	NDFD
CUTS	KIND(45)	NMD
DELT	SOLN(90)	NDFD
ELIN	KIND(45)	NMD
ERRS	SOLN(90)	NDFD
JFOR	IERROR(50,5)	NJD,5
MATD	IERROR(9,16)	NMATD,16
POTE	SOLN(90)	NDFD
REJO	IRESTR(3,50)	3,NJD
	IERROR(5,50)	5,NJD
	JNUM(50)	NJD
SEEK	DEFORM(90)	NDFD
	CURV(4095)	$\frac{NDFD}{2}$*(NDFD+1)

TABLE 3 - SUBROUTINES AND COMMON BLOCKS

SUBROUTINE	COMMON BLOCKS TO CHANGE							
	COMMON	COMMON/CONBK	COMMON/ELEMENT	COMMON/FIBER	COMMON/JOINTS	COMMON/MEMBER	COMMON/SAVEBK	COMMON/SEEKBK
ABET	X		X	X	X	X	X	X
ACIN	X		X	X	X	X		
ADYN	X		X	X	X	X	X	X
ASAN	X				X	X		X
BEAM			X	X		X		
BODY			X	X	X	X		
BOND			X	X		X		
COEN	X	X	X	X	X	X		
CONC		X	X	X		X		
CRET				X				
CUTS			X		X	X		
DEFO			X		X			
DELT			X		X	X		
ELIN			X	X		X		
ENDS						X		
ENGY			X					
ERRS					X	X		X
FAIL	X		X	X		X	X	
FORK			X			X		

SUBROUTINE	COMMON	COMMON/CONBK	COMMON/ELEMENT	COMMON/FIBER	COMMON/JOINTS	COMMON/MEMBER	COMMON/SAVEBK	COMMON/SEEKBK
FORS	X				X			
FTAB	X							
GIDE					X			X
INIT					X			
JFOR	X							
LEAF			X			X		
LINK			X	X	X	X		
LUMP					X			
MASS			X		X			
MATP				X				
MATY				X				
MEMB			X			X		
OUTS	X		X	X	X	X	X	X
PLOG			X			X	X	
POTE					X	X		X
REGO	X		X	X	X	X		X
REIN				X				
REJO					X			
SECT			X				X	
SEEK	X		X					X
STEN	X		X	X		X		
STOR	X		X					

SUBROUTINE	COMMON	COMMON/CONBK	COMMON/ELEMENT	COMMON/FIBER	COMMON/JOINTS	COMMON/MEMBER	COMMON/SAVEBK	COMMON/SEEKBK
STRN						X		
SUMY			X	X	X	X		
TABL	X							
TEST			X	X		X		
WIDE	X		X	X		X		
BLOCK DATA				X			X	
TOTAL CHANGES	16	2	29	22	21	28	7	9

TABLE 4 - SIZE OF KDATA AND DATA ARRAYS

LINEAR PROBLEM

ARRAY	NECESSARY SIZE	CURRENT VALUE
KDATA	LPI = (2*NTAB)+(5*NFF)+(2*NM)	500
DATA	$LP = (NDF*(NDF+1)/2) + \sum_{i=1}^{NTAB} 2*J(i)+4*NFF$	10000

NDF = Number of degrees of freedom
NFF = Number of joint forcing functions
NM = Number of members
NTAB = Number of forcing function tables
J(i) = Number of time-force pairs

NONLINEAR PROBLEM

ARRAY	NECESSARY SIZE	CURRENT VALUE
KDATA	LPI	500
DATA	LP+NMC*[40*NG+464]+NMWF*344	10000

NG = Number of groups of reinforcing steel in member
NMC = Number of R/C members
NMWF = Number of WF members

TYPICAL PROBLEMS

Problem A

10 members
40 degrees of freedom maximum
2 forcing functions
6 time-force pairs for each function
8 joints loaded
All R/C members with three groups of reinforcing steel

Problem B

50 members
200 degrees of freedom
10 forcing functions
6 time-force pains for each
20 joints loaded
All R/C members with three groups of reinforcing steel

Linear Problem	KDATA SIZE	DATA SIZE
Problem A	64	876
Problem B	220	20300
Nonlinear Problem		
Problem A	64	6716
Problem B	220	49500

SECTION IV

NEW SUBROUTINE DOCUMENTATION

Three new subroutines were developed and one was significantly altered for efficiency and improved accuracy. Subroutine COEN was modified by removing a segment of computations which could be more efficiently done in a separate subroutine during the input sequence. The new subroutine, CONC, is described in this section as well as the revised organization of COEN. The description of subroutine COEN is available in the Program Document of a previous contract report [4]. In addition, two new subroutines were written to replace DEFO and STRN as required by the details of the new finite element model (for development details, see section V, Volume I, of this report). The description details for these two subroutines are basically the same as the original documentation [4]; only the computational details have been changed. The source listings for both DEFO and STRN are included in this section since they are currently introduced into the program at the option of the user, and they are not included in the complete program listing in the Appendix of this volume.

1. Name of Subroutine - CONC(M)

2. Description of Subroutine

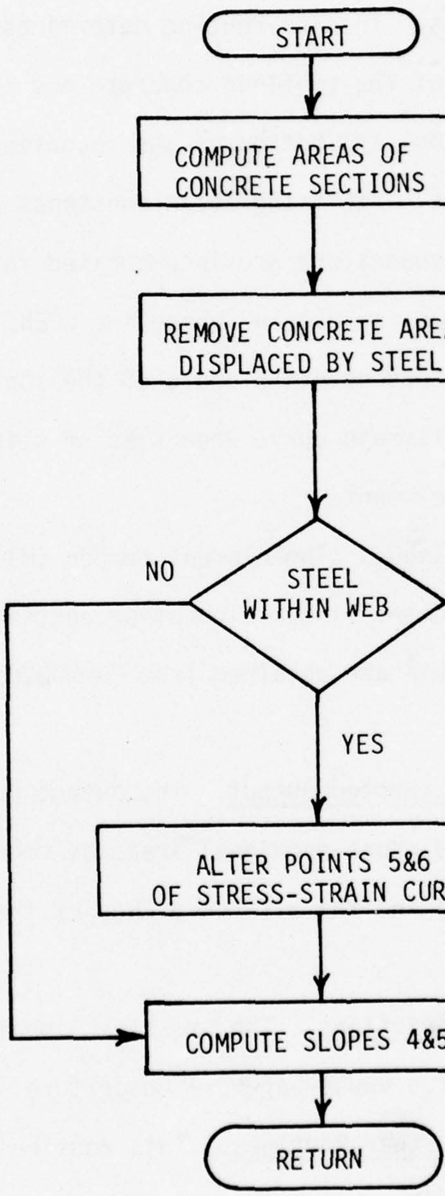
   a. Purpose and Uses. This subroutine determines the cross-sectional area of the confined concrete and the areas of the sides, top, and bottom of the unconfined concrete for each element. The integration constants of the Gauss-Legendre quadrature are incorporated into these area computations for use in subroutine COEN. This routine also alters points 5 and 6 of the confined concrete stress-strain curve when ties or stirrups are within the element.

   b. Description of Input. The element number (M) is input through the argument list. Element properties and stress-strain data are obtained from COMMON data storage.

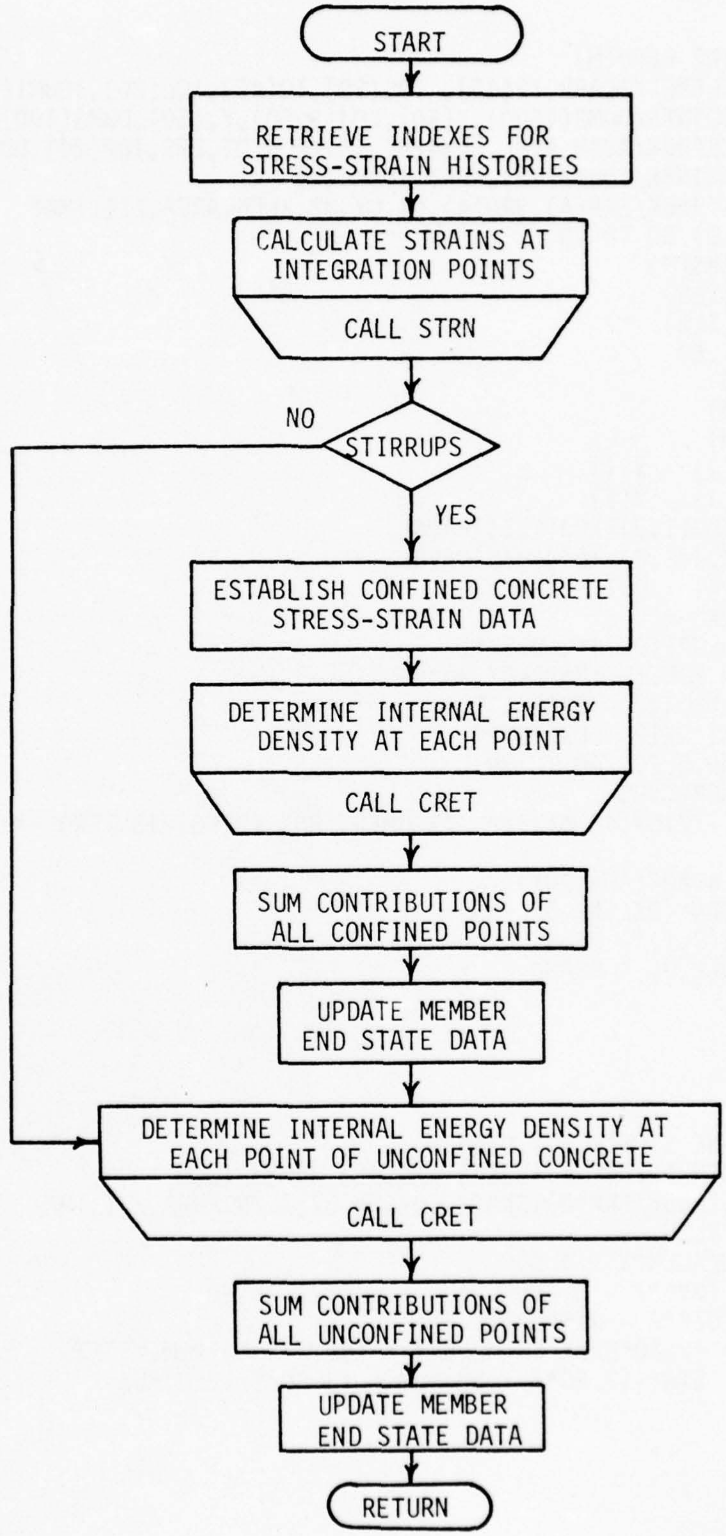
   c. Description of Expected Output. The output quantities are the factored cross-sectional areas of concrete for each element (M) and the altered points of the stress-strain curves.

   d. Limitations/Restrictions. The cross-sectional areas are factored for a Gauss-Legendre quadrature.

   e. Relationship to Other Routines. This routine is called by subroutine ACIN.


START
COMPUTE AREAS OF CONCRETE SECTIONS
REMOVE CONCRETE AREA DISPLACED BY STEEL
STEEL WITHIN WEB
NO
YES
ALTER POINTS 5&6 OF STRESS-STRAIN CURVE
COMPUTE SLOPES 4&5
RETURN

SUBROUTINE COEN



```
      SUBROUTINE DEFO(M)
      COMMON/ELEMET/ICARD,IP(45), IPL(20),IQ(45),IQL(20),IDUM1(1125)
      COMMON/JOINTS/DUM2(1500),X(50),XDJ(3,50),Y,(50),DUM3(480)
      COMMON/LEADBK/AVDM,AVGL,DUM4(5),ID1(20),DT,EPS,ID2(20),DUM5(11)
      COMMON/MAINBK/IDUM2(42),NPRT,IDUM3(6)
      COMMON/STRNBK/SRP(4),SRQ(4),UX,UY,UZ,XLEN,AREA,ZZI,IMAT
      IF(M.GT.0) GO TO 10
      ILS = IABS(M)
      I = IPL(ILS)
      J = IQL(ILS)
      XLEN = 1.E0
      GO TO 20
10    I = IP(M)
      J = IQ(M)
20    DX1 = X(J) - X(I)
      DX2 = Y(J) - Y(I)
      DU1 = (XDJ(1,J)-XDJ(1,I))*AVGL
      DU2 = (XDJ(2,J)-XDJ(2,I))*AVGL
      DX1P = DX1 + DU1
      DX2P = DX2 + DU2
      DXLEN = SORT(DX1**2+DX2**2)
      DXLENP = SORT(DX1P**2+DX2P**2)
      ADX = DX1*DX2P - DX2*DX1P
      BDX = DX1*DX1P + DX2*DX2P
      IF(BDX.NE.0.E0) GO TO 30
      WRITE (NPRT,40)
40    FORMAT (///10X,43H******  ARGUMENT BDX (DEFO) IS ZERO  ******//)
      RETURN
30    RBANG = ATAN2(ADX,BDX)
      UX = 0.5E0*(DXLENP-DXLEN)
      UY = XDJ(3,I) - RBANG
      UZ = XDJ(3,J) - RBANG
      RETURN
      END


      SUBROUTINE STRN(M,X,Y,STRAIN)
      COMMON/MEMBER/DUMMY1(5365),XDM(45),DUMMY2(360)
      COMMON/STRNBK/SRP(4),SRQ(4),UX,UY,UZ,XLEN,AREA,ZZI,IMAT
      ETA = Y/XLEN
      T = 2.E0/XLEN*X - 1.E0
      ALPHA = (UY**2 - 0.5E0*UY*UZ ÷ UZ**2)/15.E0
      BETA = (UZ**2 - UY**2)/16.E0
      STRAIN = (2.E0*UX/XLEN+ALPHA) + (BETA-4.E0*XDM(M))*T
     1        - ETA*((3.E0*T-1.E0)*UY + (3.E0*T+1.E0)*UZ)
      RETURN
      END
```

REFERENCES

1. Melosh, R. J., et al., SINGER: A Computer Code for General Analysis of Two-Dimensional Concrete Structures, AFWL-TR-74-228, Vol. III, Air Force Weapons Laboratory, Kirtland Air Force Base, NM, May 1975.

2. Holzer, S. M., et al., SINGER: A Computer Code for General Analysis of Two-Dimensional Concrete Structures, AFWL-TR-74-228, Vol. I, Air Force Weapons Laboratory, Kirtland Air Force Base, NM, May 1975.

3. Bradshaw, J. C., Nonlinear Analysis of Plane Frames, Master Thesis, Virginia Polytechnic Institute and State University, May 1975.

4. Barker, R. M., et al., SINGER: A Computer Code for General Analysis of Two-Dimensional Concrete Structures, AFWL-TR-74-228, Vol. II, Air Force Weapons Laboratory, Kirtland Air Force Base, NM, May 1975.

APPENDIX

PROGRAM SOURCE LISTING

The complete listing of the SINGER code is included in this section with the exception of the two new subroutines DEFO and STRN developed to implement the logic of the new finite element model (section V, Volume I). The listing for these two subroutines is included in section IV of this volume.

```
CABET    0  10
C                                                                       ABET   0
C       AFWL VERSION OF *SINGER*, A COMPUTER CODE FOR SIMULATING        ABET  10
C      INELASTIC AND NONLINEAR GEOMETRIC EFFECTS ON REINFORCED CONCRETE ABET  20
C      BEAM-COLUMN ELEMENTS.                                            ABET  30
C                                                                       ABET  40
C      THIS VERSION IS UPDATED AS OF 6 MAY 1976.                        ABET  50
C      ORIGINAL VERSION BEGAN BY V. P. I. AND S. U., BLACKSBURG, VA.    ABET  60
C                                                                       ABET  70
C      MAIN EXECUTIVE ROUTINE                                           ABET  80
C                                                                       ABET  90
       COMMON DATA(10000),KDATA(500)                                    ABET 100
       COMMON/ELEMET/ICAPD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),      ABET 110
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),ABET 120
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)                      ABET 130
       COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9), ABET 140
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)ABET 150
       COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), ABET 160
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),      ABET 170
     1    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)     ABET 180
       COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), ABET 190
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      ABET 200
       COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,ABET 210
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,ABET 220
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, ABET 230
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  ABET 240
     4    NTIMES,NVEL,IINITD                                            ABET 250
       COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),ABET 260
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),    ABET 270
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),ABET 280
```

```
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                       ABET 290
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45),       ABET 300
     5   YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)                      ABET 310
      COMMON /PLOTBK/ ITER                                                     ABET 320
      COMMON/SAVEBK/SAVACC(3,50),SAVAXL(2,45),SAVCRV(2,45),SAVMOM(2,45)       ABET 330
     1    ,SAVSHR(2,45),SAVSRP(3,20),SAVSRQ(3,20),SAVXDJ(3,50),               ABET 340
     2    SAVVEL(3,50),SVSTRN(12,45),SVSTRS(12,45)                             ABET 350
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,                 ABET 360
     1    LTABI,NMAX,NMAXI                                                     ABET 370
      COMMON/SEEKBK/DEFOR(90),STPSIZ(90),GRAD(90),GRADI(90),DELTAG(90),       ABET 380
     1    DIRECT(90),DIAG(90),STEP(4),DSTEP(4),FVAL(4),VALUES(7),             ABET 390
     2    DISACC,SSIZE,FUNACC,FUNMIN,CRITL,CRITU,NLIN                          ABET 400
      COMMON/STRNBK/SRP(4),SRG(4),UX,UY,UZ,XLEN,AREA,ZZI,IMAT                 ABET 410
      COMMON /TIMEBK/ TCUM,INIT,KNT                                            ABET 420
C                                                                              ABET 430
      INTEGER HEAD,DHEAD                                                       ABET 440
      DIMENSION SOLN(90)                                                       ABET 450
C                                                                              ABET 460
C     CORE ALLOCATION PARAMETERS                                               ABET 470
C                                                                              ABET 480
      NDFD=90                                                                  ABET 490
      NJD=50                                                                   ABET 500
      NLD=20                                                                   ABET 510
      NMATD=9                                                                  ABET 520
      NMAX=10000                                                               ABET 530
      NMAXI=500                                                                ABET 540
      NMD=45                                                                   ABET 550
C     PERIPHERAL UNIT PARAMETERS                                               ABET 560
      NCRD=5                                                                   ABET 570
      NPLOT=10                                                                 ABET 580
```

```
      NPRT=6                                                            ABET 590
      NSAVE=11                                                          ABET 600
      NTAPE=11                                                          ABET 610
C                                                                       ABET 620
C     PRECISION CONTROL PARAMETERS                                      ABET 630
C                                                                       ABET 640
      CRITL=1.E-9                                                       ABET 650
      CRITU=1.E-3                                                       ABET 660
      EPS=1.E-14                                                        ABET 670
      FUNACC=1.E-13                                                     ABET 680
      FUNMIN=-1.E0                                                      ABET 690
      SSIZE=1.E-5                                                       ABET 700
      TINY=1.E-36                                                       ABET 710
C                                                                       ABET 720
C     OTHER CONFIGURATION PARAMETERS                                    ABET 730
C                                                                       ABET 740
      NL=66                                                             ABET 750
C                                                                       ABET 760
C     INITIALIZE ERROR COUNTERS AND PAGE NUMBER                         ABET 770
C                                                                       ABET 780
3     IERR=0                                                            ABET 790
      IFAIL=0                                                           ABET 800
      IPAGE=0                                                           ABET 810
      IREC=0                                                            ABET 820
      IYLD = 0                                                          ABET 830
      LERR=0                                                            ABET 840
C                                                                       ABET 850
C     INITIALIZE INDEXES FOR STORAGE IN DATA ARRAY                      ABET 860
C                                                                       ABET 870
      LCURV=0                                                           ABET 880
```

```
      LTAB=0                                                            ABET 890
      LTABI=0                                                           ABET 900
      LFF=0                                                             ABET 910
      LFFI=0                                                            ABET 920
      LP=0                                                              ABET 930
      LPI=0                                                             ABET 940
      LPSI=0                                                            ABET 950
      LMAX=0                                                            ABET 960
      LMAXI=0                                                           ABET 970
C                                                                       ABET 980
C     INITIALIZE CLOCKTIME PARAMETERS AND PLOT PARAMETER                ABET 990
C                                                                       ABET1000
      INIT = 0                                                          ABET1010
      ITER = 0                                                          ABET1020
      KNT = 0                                                           ABET1030
      TCUM = 0.E0                                                       ABET1040
        CALL TICS(TIMN,IGA)                                             ABET1050
C                                                                       ABET1060
C DATA INPUT FROM CARDS.                                                ABET1070
C                                                                       ABET1080
      CALL ACIN                                                         ABET1090
C                                                                       ABET1100
C TEST FOR DATA ERRORS, STORAGE ERRORS, OR END OF DATA.                 ABET1110
C                                                                       ABET1120
      IF(TPROB.EQ.-1.0E0) GO TO 50                                      ABET1130
      IF (IERR.EQ.0 .AND. LERR.EQ.0) GO TO 8                            ABET1140
      WRITE(NPRT,7)                                                     ABET1150
7     FORMAT(1H ,69H*** DATA ERRORS. ANALYSIS FOR THIS PROBLEM IS TERMINABET1160
     1ATED. (ABET). ***)                                                ABET1170
      GO TO 50                                                          ABET1180
```

```
C                                                                       ABET1190
C INITIALIZE MEMBER ENERGY.                                             ABET1200
C                                                                       ABET1210
8     DO 10 I=1,NM                                                      ABET1220
      URM(I)=0.E0                                                       ABET1230
10    UDM (I)=0.E0                                                      ABET1240
C                                                                       ABET1250
C PERFORM STATIC ANALYSIS.                                              ABET1260
C                                                                       ABET1270
      NLIN=10*NDF                                                       ABET1280
      IF (IANAL.EQ.1) GO TO 30                                          ABET1290
      CALL ASAN(SOLN)                                                   ABET1300
      IF(IERR.EQ.0.AND.LERR.EQ.0.AND.IFAIL.EQ.0) GO TO 19               ABET1310
      PRINT 12                                                          ABET1320
12    FORMAT(1H ,69H*** ERRORS IN STATIC ANALYSIS. THIS PROBLEM IS TERMIABET1330
     1NATED. (ABET) ***)                                                ABET1340
      GO TO 50                                                          ABET1350
19    WRITE(NPRT,20)                                                    ABET1360
20    FORMAT (//1H ,44HNORMAL COMPLETION OF STATIC ANALYSIS (ABET)./1H1)ABET1370
      GO TO 3                                                           ABET1380
C                                                                       ABET1390
C PERFORM DYNAMIC ANALYSIS.                                             ABET1400
C                                                                       ABET1410
30    CALL ADYN(SOLN)                                                   ABET1420
      IF(IERR.EQ.0.AND.LERR.EQ.0.AND.IFAIL.EQ.0) GO TO 39               ABET1430
      PRINT 23                                                          ABET1440
23    FORMAT(1H ,70H*** ERRORS IN DYNAMIC ANALYSIS. THIS PROBLEM IS TERMABET1450
     1INATED. (ABET) ***)                                               ABET1460
      GO TO 50                                                          ABET1470
39    WRITE(NPRT,40)                                                    ABET1480
```

```
40    FORMAT(//1H ,45HNORMAL COMPLETION OF DYNAMIC ANALYSIS (ABET)./1H1)
      GO TO 3
50    STOP
C
C ***** GLOSSARY FOR ABET *****
C
C ACC          = ACCELERATION COMPONENTS, BY DEGREE OF FREEDOM.
C AGRP         = AREA OF STEEL IN LONGITUDINAL REINFORCEMENT GROUP.
C ATIES        = EFFECTIVE AREA OF STEEL IN A TIE OR STIRRUP.
C AVDM         = AVERAGE DIMENSIONALIZING PARAMETER = AVGL * AVGA * AVGE.
C AVGL         = AVERAGE ELEMENT LENGTH.
C BET          = JERK COMPONENTS, BY DEGREE OF FREEDOM.
C BETA         = SHEAR COEFFICIENT, BY ELEMENT.
C BMEM         = ELEMENT GROSS WIDTH, BY ELEMENT.
C BPP          = CONFINED CONCRETE WIDTH, BY ELEMENT.
C BWF          = FLANGE WIDTH OF WIDE FLANGE BEAM, BY ELEMENT.
C CA           = MACRO SHEAR FAILURE CONSTANT, SURFACE INTERCEPT.
C CB           = MACRO SHEAR FAILURE CONSTANT, CURVE SLOPE.
C CC           = MACRO SHEAR FAILURE CONSTANT, UPPER BOUND.
C CD           = MACRO SHEAR FAILURE CONSTANT, UPPER BOUND AXIAL.
C D            = DISTANCE- CONCRETE TOP TO LOWER STEEL, BY ELEMENT.
C DAS          = MASS COMPONENTS, BY DEGREE OF FREEDOM.
C DATA         = IN-CORE STORAGE FILE.
C DENS         = DENSITY (SPECIFIC WEIGHT) OF MATERIAL.
C DHEAD        = INPUT BLOCK HEADING IMAGE.
C DIS          = DISPLACEMENT COMPONENTS, BY DEGREE OF FREEDOM.
C DP           = DISTANCE- CONCRETE TOP TO UPPER STEEL, BY ELEMENT.
C DPP          = CONFINED CONCRETE HEIGHT OF ELEMENT, BY ELEMENT.
C DT           = INITIAL TIME INTERVAL FOR TRANSIENT INTEGRATION.
C DWF          = WIDE FLANGE BEAM DEPTH, BY ELEMENT.
```

```
C EC        = YOUNG'S MODULUS IN COMPRESSION.                              ABET1790
C EFFL      = EFFECTIVE LENGTH OF LONGITUDINAL REBAR GROUP.                ABET1800
C EFLM      = EFFECTIVE LENGTH OF ELEMENT, BY ELEMENT.                     ABET1810
C EGSIF     = FORCE CONVERSION FACTOR FROM ENGLISH TO SI UNITS.            ABET1820
C EGSIL     = LENGTH CONVERSION FACTOR FROM ENGLISH TO SI UNITS.           ABET1830
C EGSIS     = STRESS CONVERSION FACTOR FROM ENGLISH TO SI UNITS.           ABET1840
C ELEMET    = COMMON BLOCK OF ELEMENT INTEGER DATA.                        ABET1850
C EPS       = SMALLEST RELATIVE NUMBER IN THE MANTISSA                     ABET1860
C EPSL      = STRAIN ASSOCIATED WITH MAXIMUM CONCRETE STRESS.              ABET1870
C ERJF      = JOINT FULL-TIME ERROR, BY DEGREE OF FREEDOM.                 ABET1880
C ERJH      = JOINT HALF-TIME ERROR, BY DEGREE OF FREEDOM.                 ABET1890
C ERJZ      = JOINT ZERO-TIME ERROR, BY DEGREE OF FREEDOM.                 ABET1900
C ET        = YOUNG'S MODULUS IN TENSION.                                  ABET1910
C F         = CURRENT VALUE OF THE JOINT FORCING FUNCTION, BY D.O.F.       ABET1920
C FCFY      = CRUSHING STRENGTH OF CONCRETE OR YIELD STRENGTH OF STEEL.ABET1930
C FIBER     = COMMON BLOCK OF STRESS-STRAIN CURVE DATA.                    ABET1940
C FOR       = INITIAL JOINT FORCES, BY DEGREE OF FREEDOM.                  ABET1950
C G         = SHEAR MODULUS, BY MATERIAL.                                  ABET1960
C HEAD      = PROBLEM PAGE HEADING IMAGE.                                  ABET1970
C HTOP      = DISTANCE- TOP OF CONCRETE TO REFERENCE AXIS, BY ELEMENT.     ABET1980
C HTWF      = DISTANCE- TOP OF WIDE FLANGE TO REFERENCE AXIS.              ABET1990
C HMEM      = GROSS HEIGHT OF ELEMENT, BY ELEMENT.                         ABET2000
C IANAL     = ANALYSIS TYPE FLAG. (0=STATIC,1=DYNAMIC)                     ABET2010
C ICARD     = SEQUENCE NUMBER OF ELEMENT DATA CARDS.                       ABET2020
C ICODE     = MATERIAL TYPE CODE. (0=UNCONF.CONC.,1=CONF.CONC.,2=STEEL)ABET2030
C ICURV     = CURVATURE MATRIX FLAG. (0=NO CHANGE, 1=USE IDENTITY )        ABET2040
C IDFI      = JOINT NUMBER OF DEGREE OF FREEDOM.                           ABET2050
C IDFII     = DIRECTION NUMBER OF DEGREE OF FREEDOM.                       ABET2060
C IINITD    = INDEX FOR INITIAL GUESS OF DISPLACEMENTS. (.LT.1=READ)       ABET2070
C ILIN     = ANALYSIS COMPLEXITY FLAG. (0=LINEAR SYSTEM,.NE.0,NON-LIN)     ABET2080
```

```
C IFOR       = INITIAL JOINT FORCE FLAG. (0=NONE,.NE.0 GIVEN )             ABET2090
C IERR       = NUMBER OF INPUT ERRORS.                                     ABET2100
C IPAGE      = CURRENT NUMBER OF PAGE BEING OUTPUT.                        ABET2110
C IP         = FIRST JOINT NUMBER, PER NON-LEAF SPRING ELEMENT             ABET2120
C IPL        = FIRST JOINT NUMBER, PER LEAF SPRING COMPONENT.              ABET2130
C IPLOT      = DATA RETRIEVAL FILE FLAG. (0=DO NOT WRITE RETRIEVAL FILE)ABET2140
C IPRINT     = PRINT LEVEL FLAG. (0=MINIMUM, 1=STANDARD, 2=DETAILED)       ABET2150
C IQ         = SECOND JOINT NUMBER, PER NON-LEAF SPRING ELEMENT.           ABET2160
C IQL        = SECOND JOINT NUMBER, PER LEAF SPRING COMPONENT.             ABET2170
C IREC       = NUMBER OF RECOVERABLE ERRORS.                               ABET2180
C ISTART     = SOURCE OF INPUT DATA. (0=CARD FILE, .NE. 0=OTHER FILE)      ABET2190
C ISTOP      = ERROR OVERRIDE FLAG. (0=STOP, .NE. 0 GO IF COND. ERRORS)    ABET2200
C ISTRES     = STRESS PRINT FLAG.(0=NO STRESSES,.NE. 0 PRINT STRESSES)     ABET2210
C ITAPE      = CONTINUATION DATA DUMP SITE. (0=NO DUMP,NE.0,OTHER FILE)    ABET2220
C IUNITS     = INPUT-OUTPUT DIMENSIONAL UNITS FLAG.                        ABET2230
C IYFLAG     = NEWLY YIELDED ELEMENT FLAG. (0=NONE,1=YIELD DETECTED)       ABET2240
C IYLD       = COUNTER FOR NO. OF ELEMENTS WITH MSTAT(M)=3.                ABET2250
C JOINTS     = COMMON BLOCK OF JOINT DATA.                                 ABET2260
C KDATA      = INDEXES OF RECORDS IN IN-CORE FILE CALLED DATA.             ABET2270
C LCURV      = STARTING INDEX OF CURVATURE DATA IN DATA ARRAY.             ABET2280
C LEADBK      = COMMON BLOCK OF DATA LEADING PROBLEM SOLUTION.             ABET2290
C LERR       = STORAGE ERROR FLAG. (0= NO ERRORS)                          ABET2300
C LFF        = STARTING INDEX OF JOINT FORCING FUNCTIONS IN DATA ARRAY.    ABET2310
C LFFI       = STARTING INDEX OF JOINT FORCING FUNCTIONS IN KDATA ARRAY.ABET2320
C LINE       = CURRENT NUMBER OF THE LINE BEING OUTPUT.                    ABET2330
C LMAX       = MAXIMUM NUMBER OF DATA LOCATIONS USED IN THIS PROBLEM.      ABET2340
C LMAXI      = MAXIMUM NUMBER OF KDATA LOCATIONS USED IN THIS PROBLEM.     ABET2350
C LP         = STARTING INDEX OF PLASTIC STRAIN DATA IN DATA ARRAY.        ABET2360
C LPI        = STARTING INDEX OF PLASTIC STRAIN DATA IN KDATA ARRAY.       ABET2370
C LPSI       = STARTING INDEX OF STRESS HISTORY DATA IN KDATA ARRAY.       ABET2380
```

```
C LTAB       = STARTING INDEX OF FUNCTION TABLES IN DATA ARRAY.              ABET2390
C LTABI      = STARTING INDEX OF FUNCTION TABLES IN KDATA ARRAY.             ABET2400
C MAINBK      = COMMON BLOCK FOR PRINCIPAL SIMULATION CONSTANTS.             ABET2410
C MATR       = CONFINED CONCRETE MATERIAL I.D., BY ELEMENT.                  ABET2420
C MATW       = MATERIAL OF WIDE FLANGE BEAM COMPONENT, BY ELEMENT.           ABET2430
C MBAR       = MATERIAL OF LONGITUDINAL REINFORCEMENT, BY ELEMENT.           ABET2440
C MCODE      = UNCONFINED CONCRETE MATERIAL I.D., BY ELEMENT.                ABET2450
C MEMBER     = COMMON BLOCK OF MEMBER PARAMETERS                             ABET2460
C MSHEAR     = ELEMENT SHEAR  FLAG. (0=IGNORE, 1=CHECK)                      ABET2470
C MSTAT      = ELEMENT STATUS FLAG, BY ELEMENT.                              ABET2480
C MTIES      = MATERIAL OF LATERAL REINFORCEMENT, BY ELEMENT.                ABET2490
C MTYPE      = TYPE OF ELEMENT.                                              ABET2500
C NACC       = NUMBER OF INITIAL ACCELERATIONS SPECIFIED BY USER.            ABET2510
C NAME       = NAMES OF MATERIALS FOR THIS PROBLEM.                          ABET2520
C NCRD       = NUMBER OF PERIPHERAL FILE FOR CARD DATA.                      ABET2530
C NDF        = NUMBER OF DEGREES OF FREEDOM FOR THIS PROBLEM.                ABET2540
C NDFD       = MAXIMUM NUMBER OF DEGREES OF FREEDOM MAPPED.                  ABET2550
C NDFJ       = NUMBER OF DEGREES OF FREEDOM AT JOINTS.                       ABET2560
C NDIS       = NUMBER OF INITIAL DISPLACEMENTS SPECIFIED BY USER.            ABET2570
C NDL        = NUMBER OF DISTRIBUTED LOADINGS GIVEN BY USER.                 ABET2580
C NFF        = NUMBER OF FORCING FUNCTIONS SPECIFIED BY USER.                ABET2590
C NGRP       = NUMBER OF GROUPS OF LONGITUDINAL REINFORCEMENT, BY ELEM.      ABET2600
C NINC       = NUMBER OF INITIAL CONDITIONS SPECIFIED BY USER (TOTAL)        ABET2610
C NJ         = NUMBER OF JOINTS FOR THIS PROBLEM.                            ABET2620
C NJD        = MAXIMUM NUMBER OF  JOINTS MAPPED.                             ABET2630
C NJER       = NUMBER OF CONDITIONAL STOP ERRORS.                            ABET2640
C NJOR       = NUMBER OF JOINTS WITH POINT LOADINGS GIVEN BY USER.           ABET2650
C NL         = NUMBER OF LINES THAT MAY BE PRINTED ON AN OUTPUT PAGE.        ABET2660
C NLD        = MAXIMUM NUMBER OF LEAF SPRINGS MAPPED.                        ABET2670
C NLS        = NUMBER OF LEAF SPRINGS FOR THIS PROBLEM.                      ABET2680
```

```
C NLSR     = NUMBER OF LEAF SPRING RIGIDITY CONSTRAINTS.                  ABET2690
C NM       = NUMBER OF ELEMENTS FOR THIS PROBLEM.                         ABET2700
C NMAS     = NUMBER OF MASSES FOR THIS PROBLEM. (INCLUDES BODY MASS)      ABET2710
C NMAT     = NUMBER OF MATERIALS FOR THIS PROBLEM.                        ABET2720
C NMATD    = MAXIMUM NUMBER OF MATERIALS MAPPED.                          ABET2730
C NMAX     = MAXIMUM INDEX OF  DATA ARRAY.                                ABET2740
C NMAXI    = MAXIMUM INDEX OF KDATA ARRAY.                                ABET2750
C NMD      = MAXIMUM NUMBER OF ELEMENTS MAPPED                            ABET2760
C NPLOT    = NUMBER OF PERIPHERAL UNIT FOR RETRIEVAL FILE DATA.           ABET2770
C NPRT     = NUMBER OF PERIPHERAL UNIT FOR PRINT DATA.                    ABET2780
C NSAVE    = NUMBER OF PERIPHERAL UNIT FOR CONTINUATION DATA. (WRITES)ABET2790
C NSPAC    = NUMBER OF TIMES A PARTICULAR STIRRUP SPACING OCCURS.         ABET2800
C NTAB     = NUMBER OF FUNCTION TABLES SPECIFIED BY USER.                 ABET2810
C NTAPE    = NUMBER OF PERIPHERAL UNIT FOR CONTINUATION DATA. (READS)     ABET2820
C NTIES    = NUMBER OF LATERAL REINFORCEMENT GROUPS, BY ELEMENT.          ABET2830
C NTIMES   = MAXIMUM NUMBER OF TIME EVALS FOR THIS PROBLEM.               ABET2840
C NVEL     = NUMBER OF INITIAL VELOCITIES SPECIFIED BY USER.              ABET2850
C PDP      = CONCRETE CONFINEMENT FACTOR, BY ELEMENT.                     ABET2860
C PI       = 3.14159265358978D0.                                          ABET2870
C PR       = POISSON'S RATIO, BY MATERIAL.                                ABET2880
C RERF     = MAXIMUM OF JOINT FULL-TIME ERRORS.                           ABET2890
C RERH     = MAXIMUM OF JOINT HALF-TIME ERRORS.                           ABET2900
C RERZ     = MAXIMUM OF JOINT ZERO-TIME ERRORS.                           ABET2910
C S        = STRESS DATA ARRAY FOR CURRENTLY ACTIVE POINT.                ABET2920
C SCALE    = COMMON BLOCK OF SCALE FACTORS FOR DIMENSIONAL UNITS.         ABET2930
C SERR     = MAXIMUM TOLERABLE RELATIVE ERROR IN THE TOTAL ENERGY.        ABET2940
C SLOPE    = SLOPES OF SEGMENTS OF STRESS-STRAIN CURVES.                  ABET2950
C SPRING   = LEAF SPRING FLEXIBILITY COEFFICIENTS, BY LEAF.               ABET2960
C ST       = STRESS-STRAIN DATA FOR DEFAULT STEELS.                       ABET2970
C STIES    = SPACING OF TIES OR STIRRUPS.                                 ABET2980
```

```
C STN         = STRAIN COORDINATES OF MATERIALS FOR THIS PROBLEM.        ABET2990
C STORE       = COMMON BLOCK OF STORAGE INDEXES.                         ABET3000
C STS         = STRESS COORDINATES OF MATERIALS FOR THIS PROBLEM.        ABET3010
C TBEGIN      = TIME AT WHICH TRANSIENT INTEGRATION BEGINS.              ABET3020
C TFWF        = THICKNESS- FLANGE OF WIDE FLANGE BEAM, BY ELEMENT.       ABET3030
C THALT       = TIME AT WHICH TRANSIENT INTEGRATION ENDS.                ABET3040
C TIME        = CURRENT TIME OF TRANSIENT INTEGRATION.                   ABET3050
C TINK        = APPROXIMATE TRANSIENT TIME FOR PRINTING STATE DATA.      ABET3060
C TINY        = THE SMALLEST NUMBER THE COMPUTER RECOGNIZES.             ABET3070
C TPROB       = MAXIMUM TIME PERMITTED FOR THIS COMPUTER RUN.            ABET3080
C TWWF        = THICKNESS- WEB OF WIDE FLANGE BEAM, BY ELEMENT.          ABET3090
C UNLK        = UNLOADING CURVE CONSTANT FOR CONCRETE.                   ABET3100
C VEL         = VELOCITY COMPONENTS, BY DEGREE OF FREEDOM.               ABET3110
C WFPI        = PRINCIPAL MOMENT OF INERTIA OF WIDE FLANGE BEAM.         ABET3120
C X           = X COORDINATE, BY JOINT.                                  ABET3130
C XBEG        = DISTANCE- JOINT +P+ TO START OF LONGITUDINAL REBAR GROUP.ABET3140
C XBEGM       = DISTANCE- JOINT +P+ TO EFFECTIVE START OF ELEMENT.       ABET3150
C XBEGS       = DISTANCE- JOINT +P+ TO START OF STIRRUPS.                ABET3160
C XDJ         = CURRENT VALUE OF GLOBAL JOINT DISPLACEMENTS, BY D.O.F.   ABET3170
C XL          = ELEMENT LENGTH, BY ELEMENT.                              ABET3180
C Y           = Y COORDINATE, BY JOINT.                                  ABET3190
C YBAR        = DISTANCE- REFERENCE AXIS TO LONGITUDINAL REBAR.          ABET3200
C YLDS        = YIELD STRESS OF STIRRUPS, BY ELEMENT.                    ABET3210
C ZI          = AVG. CRACKED   MOMENT OF INERTIA OF CONCRETE, BY ELEMENT.ABET3220
      END                                                                ABET3230
```

```
CACIN    0  10
      SUBROUTINE ACIN                                                   ACIN   0
C                                                                       ACIN  10
C     THIS SUBROUTINE READS INPUT DATA FROM CARDS                       ACIN  20
C                                                                       ACIN  30
C                                                                       ACIN  40
      COMMON DATA(10000),KDATA(500)                                     ACIN  50
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),       ACIN  60
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),ACIN  70
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)                      ACIN  80
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9), ACIN  90
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)ACIN 100
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), ACIN 110
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),      ACIN 120
     1    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)     ACIN 130
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), ACIN 140
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      ACIN 150
      INTEGER HEAD,DHEAD                                                ACIN 160
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,ACIN 170
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,ACIN 180
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, ACIN 190
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  ACIN 200
     4    NTIMES,NVEL,IINITD                                            ACIN 210
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),ACIN 220
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),    ACIN 230
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),ACIN 240
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                ACIN 250
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45), ACIN 260
     5   YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)               ACIN 270
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,          ACIN 280
```

```
     1    LTABI,NMAX,NMAXI                                              ACIN 290
C                                                                       ACIN 300
C                                                                       ACIN 310
C     READ CONTROL DATA, BLOCK 1.                                       ACIN 320
C                                                                       ACIN 330
      CALL GIDE                                                         ACIN 340
      IF(TPROB.EQ.-1.0E0) GO TO 250                                     ACIN 350
      IF (ISTART.EQ.0) GO TO 30                                         ACIN 360
C                                                                       ACIN 370
C     READ RECOVERY DATA.                                               ACIN 380
      CALL REGO(ISTART)                                                 ACIN 390
      GO TO 70                                                          ACIN 400
C                                                                       ACIN 410
C     READ JOINT COORDINATES AND CONSTRAINTS, BLOCK 2.                  ACIN 420
C                                                                       ACIN 430
   30 CALL REJO                                                         ACIN 440
C                                                                       ACIN 450
C     READ MATERIAL TABLE DATA, BLOCK 3.                                ACIN 460
C                                                                       ACIN 470
      CALL MATP                                                         ACIN 480
C                                                                       ACIN 490
C     READ ELEMENT DATA, BLOCK 4.                                       ACIN 500
C                                                                       ACIN 510
      CALL BEAM                                                         ACIN 520
      CALL LINK                                                         ACIN 530
C                                                                       ACIN 540
C     READ LUMPED MASS DATA, BLOCK 5.                                   ACIN 550
C                                                                       ACIN 560
      CALL MASS                                                         ACIN 570
      CALL BODY                                                         ACIN 580
```

```
C
C     SET NEGLIGIBLE MASSES ZERO AND MAKE COUNT.
      IF (NMAS.EQ.0) GO TO 60
      RHO=0.E0
      DO 40 J=1,NJ
      DO 40 I=1,3
40    RHO=RHO+DAS(I,J)
      IF (RHO.EQ.0.E0) GO TO 60
      I1=0
      TEST=EPS*10.E0
      DO 50 J=1,NJ
      DO 50 I=1,3
      I1=I1+1
      IF (DAS(I,J)/RHO.GT.TEST) GO TO 50
      I1=I1-1
      DAS(I,J)=0.E0
50    CONTINUE
      NMAS=I1
C
C     READ INITIAL CONDITIONS, BLOCK 6.
C
60    CALL INIT
C
C     READ FUNCTION TABLES, BLOCK 7.
C
      CALL FTAB
C
C     READ JOINT FORCING FUNCTIONS, BLOCK 8.
C
      CALL JFOR
```

```
C
C     GENERATE EXTRA ELEMENT CATA FOR INTERIOR JOINTS.
      CALL CUTS
C
C CALCULATE LOCATION OF INTEGRATION POINTS.
      CALL SECT
C
C
C     ESTABLISH INTEGRATION CONSTANTS FOR CONCRETE AND POINTS 5 AND 6
C     OF THE STRESS-STRAIN CURVE.
C
      DO 65 M=1,NM
      IF (MTYPE(M).LT.4) CALL CONC(M)
65    CONTINUE
C     PRINT PROBLEM DESCRIPTIVE SUMMARY
70    CALL SUMY
C
C     NOTE ERROR STATUS.
C
      IF (IERR.EQ.0) GO TO 90
      PRINT 80, IERR
      GO TO 250
80    FORMAT (//1H ,38H *** NUMBER OF INPUT ERRORS DETECTED =,I5,33H , P
     1ROBLEM TERMINATED (ACIN). ***)
C
C     INITIALIZE STORAGE ADDRESS FOR PLASTIC STRAIN AND STRESS DATA.
C
90    LMAX=LP
      LPSI=LPI+NM
      LMAXI=LPSI+NM
```

```
      IF (LMAXI.LE.NMAXI) GO TO 110
      PRINT 100
100   FORMAT (//1H ,63H*** PRCBLEM STORAGE REQUIREMENTS EXCEED KDATA ARRA
     1AY (ACIN). ***)
      LERR=1
      GO TO 140
110   NM2=2*NM
      DO 120 M=1,NM2
      K=LPI+M
120   KDATA(K)=0
C
C     ESTABLISH DATA STORAGE FOR PLASTIC STRESS-STRAIN HISTORY.
C
      DO 130 M=1,NM
      IF(MSTAT(M).EQ.1) IYLD = IYLD + 1
      IF(MSTAT(M).EQ.3) CALL STOR(M)
130   CONTINUE
C
C     PRINT STORAGE LOCATION INDEXES.
C
140   IF (LERR.NE.0) GO TO 190
      WRITE (NPRT,150)
150   FORMAT (///10X,46HSTORAGE LOCATION INDEXES IN DATA ARRAY (ACIN)./)
      WRITE (NPRT,160) LCURV,LTAB,LFF,LP,LMAX,NMAX
160   FORMAT (14X,17HMINIMIZATION DATA,I19/14X,15HFUNCTION TABLES,I21/14
     1X,17HFORCING FUNCTIONS,I19/14X,19HPLASTIC STRAIN DATA,I17/14X,12HE
     2ND OF ARRAY,I24/14X,16HSPACES ALLOCATED,I20//)
      WRITE (NPRT,170)
170   FORMAT (10X,47HSTORAGE LOCATION INDEXES IN KDATA ARRAY (ACIN)./)
      WRITE (NPRT,180) LTABI,LFFI,LPI,LPSI,LMAXI,NMAXI
```

```
180    FORMAT (14X,15HFUNCTION TABLES,I21/14X,17HFORCING FUNCTIONS,I19/14ACIN1490
      1X,19HPLASTIC STRAIN DATA,I17/14X,19HSTRESS HISTORY DATA,I17/14X,12ACIN1500
      2HEND OF ARRAY,I24/14X,16HSPACES ALLOCATED,I20)                   ACIN1510
       GO TO 210                                                         ACIN1520
C                                                                        ACIN1530
190    PRINT 200                                                         ACIN1540
200    FORMAT (//1H ,110H*** PREVIOUS DATA ERROR, CALCULATION OF STORAGE ACIN1550
      1INDEXES FOR DATA AND KDATA ARRAYS IS NOT COMPLETED (ACIN). ***)   ACIN1560
       GO TO 250                                                         ACIN1570
C                                                                        ACIN1580
C CALCULATE DIMENSIONALIZING PARAMETERS.                                 ACIN1590
C                                                                        ACIN1600
210    AVGL=0.E0                                                         ACIN1610
       AVGA=0.E0                                                         ACIN1620
       DO 230 M=1,NM                                                     ACIN1630
       AVGL=AVGL+XL(M)                                                   ACIN1640
       IF (MTYPE(M).EQ.4) GO TO 220                                      ACIN1650
       AVGA=AVGA+HMEM(M)*BMEM(M)                                         ACIN1660
       GO TO 230                                                         ACIN1670
220    AVGA=AVGA+2.E0*BWF(M)*TFWF(M)+TWWF(M)*(HTWF(M)-2.E0*TFWF(M))      ACIN1680
230    CONTINUE                                                          ACIN1690
       AVGL=AVGL/FLOAT(NM)                                               ACIN1700
       AVGA=AVGA/FLOAT(NM)                                               ACIN1710
       AVGE=0.E0                                                         ACIN1720
       DO 240 I=1,NMAT                                                   ACIN1730
240    AVGE=AVGE+ET(I)+EC(I)                                             ACIN1740
       AVGE=AVGE/(2.E0*FLOAT(NMAT))                                      ACIN1750
       AVDM=AVGL*AVGA*AVGE                                               ACIN1760
C                                                                        ACIN1770
250    RETURN                                                            ACIN1760
```

```
C                                                                   ACIN1790
      END                                                           ACIN1800
```

```
CADYN    0  10
      SUBROUTINE ADYN(SOLN)                                             ADYN   0
C                                                                       ADYN  10
      COMMON DATA(10000),KDATA(500)                                     ADYN  20
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),       ADYN  30
     1    MATH(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),ADYN  40
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)                      ADYN  50
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9), ADYN  60
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)ADYN  70
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), ADYN  80
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),      ADYN  90
     1    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)     ADYN 100
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), ADYN 110
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      ADYN 120
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,ADYN 130
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,ADYN 140
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, ADYN 150
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  ADYN 160
     4    NTIMES,NVEL,IINITD                                            ADYN 170
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),ADYN 180
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),    ADYN 190
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),ADYN 200
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                ADYN 210
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45), ADYN 220
     5   YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)               ADYN 230
      COMMON/SAVEBK/SAVACC(3,50),SAVAXL(2,45),SAVCRV(2,45),SAVMOM(2,45) ADYN 240
     1    ,SAVSHR(2,45),SAVSRP(3,20),SAVSRQ(3,20),SAVXDJ(3,50),         ADYN 250
     2    SAVVEL(3,50),SVSTRN(12,45),SVSTRS(12,45)                      ADYN 260
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,          ADYN 270
     1    LTABI,NMAX,NMAXI                                              ADYN 280
```

```
      COMMON/SEEKBK/DEFOR(90),STPSIZ(90),GRAD(90),GRADI(90),DELTAG(90),
     1     DIRECT(90),DIAG(90),STEP(4),DSTEP(4),FVAL(4),VALUES(7),
     2     DISACC,SSIZE,FUNACC,FUNMIN,CRITL,CRITU,NLIN
      COMMON/STRNBK/SRP(4),SRQ(4),UX,UY,UZ,XLEN,AREA,ZZI,IMAT
C
      INTEGER HEAD,DHEAD
      DIMENSION SOLN(90)
      DATA DTMIN/1.E-07/
C
C  THIS SUBROUTINE PERFORMS DYNAMIC ANALYSIS
C
C INITIALIZE PROBLEM DATA WITH STATIC ANALYSIS
      CALL ASAN(SOLN)
C
C ADD CALCULATED INITIAL DISPLACEMENTS TO PRESCRIBED DISPLACEMENTS.
C
      DO 151 I=1,NDFJ
      J=IDFI(I)
      K=IDFII(I)
151   DIS(K,J)=DIS(K,J)+SOLN(I)
      L=NDFJ+1
      DO 152 I=L,NDF
       M=IDFI(I)
152   DISM(M)=DISM(M)+SOLN(I)
C
C SET INITIAL DISPLACEMENTS FOR DYNAMIC ANALYSIS
C
      DO 156 I=1,NDFJ
      J=IDFI (I)
      K=IDFII(I)
```

```
      SOLN(I)=DIS(K,J)                                                  ADYN 590
156   XDJ(K,J)=SOLN(I)                                                  ADYN 600
      L=NDFJ+1                                                          ADYN 610
      DO 157 I=L,NDF                                                    ADYN 620
      M=IDFI(I)                                                         ADYN 630
      SOLN(I)=DISM(M)                                                   ADYN 640
157   XDM(M)=SOLN(I)                                                    ADYN 650
C                                                                       ADYN 660
C CALCULATE INITIAL ACCELERATIONS.                                      ADYN 670
C                                                                       ADYN 680
      IF(NMAS.EQ.0) GO TO 170                                           ADYN 690
      IF(IFOR.EQ.0.AND.NFF.EQ.0) GO TO 170                              ADYN 700
      DO 161  J=1,NJ                                                    ADYN 710
      DO 161  K=1,3                                                     ADYN 720
161   F(K,J)=0.E0                                                       ADYN 730
      IF(NFF.EQ.0) GO TO 162                                            ADYN 740
      CALL FORS(TIME)                                                   ADYN 750
162    DO 164 J=1,NJ                                                    ADYN 760
      DO 164 K=1,3                                                      ADYN 770
      DFOR=F(K,J)-FOR(K,J)                                              ADYN 780
      IF(DAS(K,J).EQ.0.E0) GO TO 164                                    ADYN 790
      DO 163 I=1,NDFJ                                                   ADYN 800
      IF (J.EQ.IDFI(I).AND.K.EQ.IDFII(I)) GO TO 165                     ADYN 810
163   CONTINUE                                                          ADYN 820
      GO TO 164                                                         ADYN 830
165   ACC(K,J) = DFOR/DAS(K,J) + ACC(K,J)                               ADYN 840
164   CONTINUE                                                          ADYN 850
C                                                                       ADYN 860
C FOR MASSLESS DEGREES-OF-FREEDOM SET VELOCITY, ACCELERATION,           ADYN 870
C AND BETA EQUAL TO ZERO                                                ADYN 880
```

```
C                                                                       ADYN 890
170   DO 172 J=1,NJ                                                     ADYN 900
      DO 172 K=1,3                                                      ADYN 910
      IF(DAS(K,J ).NE.0.E0) GO TO 172                                   ADYN 920
      VEL(K,J)=0.E0                                                     ADYN 930
      ACC(K,J)=0.E0                                                     ADYN 940
      BET(K,J)=0.E0                                                     ADYN 950
172    CONTINUE                                                         ADYN 960
C                                                                       ADYN 970
      IFOR=0                                                            ADYN 980
      IF(NFF.NE.0) IFOR=1                                               ADYN 990
      DO 201 J=1,NJ                                                     ADYN1000
      DO 201 K=1,3                                                      ADYN1010
      F(K,J)=0.E0                                                       ADYN1020
201   FOR(K,J)=0.E0                                                     ADYN1030
C BEGIN CURRENT TIME STEP AND UPDATE ERROR DATA                         ADYN1040
C                                                                       ADYN1050
203   RERZ=RERF                                                         ADYN1060
      DO 206 J=1,NJ                                                     ADYN1070
      DO 206 K=1,3                                                      ADYN1080
206   ERJZ(K,J)=ERJF(K,J)                                               ADYN1090
C     INCREMENT TIME                                                    ADYN1100
209   TIME=TIME+DT                                                      ADYN1110
C                                                                       ADYN1120
C CHECK FOR COMPLETION OF INTEGRATION                                   ADYN1130
C                                                                       ADYN1140
      CALL TICS(TIMN,IGA)                                               ADYN1150
      IF(IGA.EQ.2) GO TO 400                                            ADYN1160
      WRITE(NPRT,971) TIME,DT                                           ADYN1170
971   FORMAT(1H1,17HSOLUTION FOR TIME,0PE13.5,23H, WITH TIME INTERVAL OFADYN1180
```

```
     1,E13.5//)                                                          ADYN1190
C                                                                        ADYN1200
C                                                                        ADYN1210
C PREDICT DISPLACEMENTS FOR CURRENT TIME INTERVAL AND FORM 1D            ADYN1220
C DISPLACEMENT ARRAY                                                     ADYN1230
C                                                                        ADYN1240
      IF (NMAS.EQ.0) GO TO 220                                           ADYN1250
      DO 211 I=1,NDFJ                                                    ADYN1260
      J=IDFI(I)                                                          ADYN1270
      K=IDFII(I)                                                         ADYN1280
211   SOLN(I)=BET(K,J)*DT**3/6.E0+ACC(K,J)*DT**2/2.E0+VEL(K,J)*DT+       ADYN1290
     1      DIS(K,J)                                                     ADYN1300
      L=NDFJ+1                                                           ADYN1310
      DO 212 I=L,NDF                                                     ADYN1320
      M=IDFI(I)                                                          ADYN1330
212   SOLN(I)=DISM(M)                                                    ADYN1340
C CALCULATE FORCING FUNCTION FOR CURRENT TIME STEP                       ADYN1350
C                                                                        ADYN1360
220   IF(NFF.EQ.0) GO TO 240                                             ADYN1370
      CALL FORS(TIME)                                                    ADYN1380
      DO 221 I=1,NFF                                                     ADYN1390
      L=LFFI+5*I                                                         ADYN1400
      J=KDATA(L-4)                                                       ADYN1410
      K=KDATA(L-2)                                                       ADYN1420
221   FOR(K,J)=F(K,J)                                                    ADYN1430
C                                                                        ADYN1440
C SOLVE FOR DISPLACEMENTS BY FUNCTION MINIMIZATION PROCEDURE             ADYN1450
C                                                                        ADYN1460
240   CALL SEEK(SOLN,VALUEM)                                             ADYN1470
      DO 243 I=1,NDFJ                                                    ADYN1480
```

```
      J=IDFI(I)
      K=IDFII(I)
243   XDJ(K,J)=SOLN(I)
      L=NDFJ+1
      DO 244 I=L,NDF
      M=IDFI(I)
244   XDM(M)=SOLN(I)
C
C     CHECK STATUS OF ELASTIC-PLASTIC MEMBERS.
      IYFLAG = 0
      IF (IYLO.LT.NM) CALL TEST (IYFLAG)
C
C STORE ENERGIES AND STRESS RESULTANTS, AND CHECK MEMBER FAILURE
C     CRITERIA.
      CALL FAIL
C
C     REPEAT DYNAMIC CALCULATION IF MEMBERS HAVE YIELDED.
      IF (IYFLAG.EQ.1.AND.LERR.EQ.0.AND.IERR.EQ.0) GO TO 240
C
C CALCULATE ERROR MEASURES FOR CURRENT TIME STEP
C
      WRITE(NPRT,971) TIME,DT
      CALL ERRS(SOLN,VALUEM)
      WRITE(NPRT,972)RERZ,RERH,RERF
972   FORMAT(1H0,5X,39HTOTAL ERROR MEASURES, ZERO-TIME ERROR=,0PE13.5/
     11H ,28X,16HHALF-TIME ERROR=,E14.5/1H ,28X,16HFULL-TIME ERROR=,E14.
     25//)
C
C IF DATA ERRORS HAVE OCCURED, SAVE PREVIOUS SOLUTION ON TAPE FOR
C SUBSEQUENT RESTART.
```

```
      IF(IERR.NE.0.OR.LERR.NE.0) GO TO 400
C
C CHECK SOLUTION ACCURACY.  IF NECESSARY, REDUCE TIME INTERVAL
C AND REPEAT TIME STEP.
C
      IF (CRITU.EQ.0.E0) GO TO 289
      IF(RERF.LE.CRITU) GO TO 283
      ERRXX=RERF
      GO TO 284
283   ERRXX= RERH-(RERZ+RERF)/2.
      IF(ERRXX.LE.CRITU) GO TO 289
284   WRITE(NPRT,973) ERRXX,CRITU
973   FORMAT(1H0,5X,46HTHE REQUIRED ACCURACY WAS NOT ACHIEVED (ERROR=,1P
     1E13.5,11H, REQUIRED=,E13.5,1H) )
      IF(DT.EQ.DTMIN) GO TO 288
      TIME=TIME-DT
      DT=0.6*DT
      IF (DT.LT.DTMIN) DT = DTMIN
      WRITE (NPRT,974)
974   FORMAT(1H ,5X,65HTHE TIME INTERVAL IS REDUCED AND THE INTEGRATION
     1STEP IS REPEATED)
      GO TO 209
288   WRITE(NPRT,975)
975   FORMAT (1H ,80H**** WITH MINIMUM TIME INTERVAL. RESULTS ARE REPORT
     1ED AND INTEGRATION CONTINUES.//)
289   CONTINUE
C
C INTEGRATION FOR CURRENT TIME STEP IS COMPLETE.
C UPDATE RESPONSE RESULTS.
C
```

```
      IF (NMAS.EQ.0)  GO TO 295                                         ADYN2090
      DO 291  I=1,NDFJ                                                  ADYN2100
      J=IDFI(I)                                                         ADYN2110
      K=IDFII(I)                                                        ADYN2120
      IF(DAS(K,J).EQ.0.E0)  GO TO 291                                   ADYN2130
      BET(K,J)=6.E0*((SOLN(I)-DIS(K,J))/(DT*DT)-VEL(K,J)/DT             ADYN2140
     1 -ACC(K,J)*0.5E0)/DT                                              ADYN2150
      VEL(K,J)=VEL(K,J)+ACC(K,J)*DT+BET(K,J)*DT*DT*0.5E0                ADYN2160
      ACC(K,J)=ACC(K,J)+BET(K,J)*DT                                     ADYN2170
  291 DIS(K,J)=SOLN(I)                                                  ADYN2180
      GO TO 297                                                         ADYN2190
  295 DO 296 I=1,NDFJ                                                   ADYN2200
      J=IDFI(I)                                                         ADYN2210
      K=IDFII(I)                                                        ADYN2220
  296 DIS(K,J)=SOLN(I)                                                  ADYN2230
  297 L=NDFJ+1                                                          ADYN2240
      DO 298 I=L,NDF                                                    ADYN2250
      M=IDFI(I)                                                         ADYN2260
  298 DISM(M)=SOLN(I)                                                   ADYN2270
C                                                                       ADYN2280
C PRINT RESPONSE RESULTS                                                ADYN2290
C                                                                       ADYN2300
      CALL OUTS                                                         ADYN2310
      IF(IPLOT.EQ.1)  CALL PLOG(NPLOT)                                  ADYN2320
      IF(IFAIL.EQ.1)  RETURN                                            ADYN2330
C                                                                       ADYN2340
C DETERMINE TIME INTERVAL FOR NEXT TIME STEP                            ADYN2350
C                                                                       ADYN2360
      IF (CRITU.EQ.0.E0)  GO TO 382                                     ADYN2370
      IF(ERRXX.GT.CRITL)  GO TO 381                                     ADYN2380
```

```
      WRITE(NPRT,990) ERRXX                                             ADYN2390
990   FORMAT(1H0,41HTIME INTERVAL IS INCREASED FOR NEXT STEP.,9H ERROR  ADYN2400
     1=0PE12.4)                                                         ADYN2410
      DT=1.85*DT                                                        ADYN2420
      GO TO 382                                                         ADYN2430
381    WRITE(NPRT,991)ERRXX                                             ADYN2440
991   FORMAT(1H0,43HTIME INTERVAL IS NOT CHANGED FOR NEXT STEP.,9H   ERRADYN2450
     1OR=,0PE12.3)                                                      ADYN2460
382    CONTINUE                                                         ADYN2470
C                                                                       ADYN2480
C GO TO NEXT TIME STEP                                                  ADYN2490
C                                                                       ADYN2500
      IF(IERR.EQ.0.AND.LERR.EQ.0) GO TO 203                             ADYN2510
C                                                                       ADYN2520
C INTEGRATION FOR PROBLEM IS COMPLETE, RETURN TO MAIN.                  ADYN2530
C                                                                       ADYN2540
400   IF (ITAPE.NE.0) CALL REGO (-ITAPE)                                ADYN2550
      RETURN                                                            ADYN2560
      END                                                               ADYN2570
```

```
CASAN    0  10
      SUBROUTINE ASAN (SOLN)
C
C     THIS SUBROUTINE DIRECTS STATIC ANALYSIS
C     TO OBTAIN INITIAL PROBLEM DATA.
C
      COMMON DATA(10000),KDATA(500)
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50),
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),
     1    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20),
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,
     4    NTIMES,NVEL,IINITD
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45),
     5   YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,
     1    LTABI,NMAX,NMAXI
      COMMON/SEEKBK/DEFOR(90),STPSIZ(90),GRAD(90),GRADI(90),DELTAG(90),
     1    DIRECT(90),DIAG(90),STEP(4),DSTEP(4),FVAL(4),VALUES(7),
     2    DISACC,SSIZE,FUNACC,FUNMIN,CRITL,CRITU,NLIN
C
      INTEGER HEAD,DHEAD
```

```
      DIMENSION SOLN(90)
C
C     SET INITIAL VALUES.
C
      DT=TINK
      TIME=TBEGIN
      ICURV=1
      DO 10 J=1,NJ
      DO 10 K=1,3
      F(K,J)  =  0.E0
      XDJ(K,J)=0.E0
      DER(K,J)=0.E0
      RESENG(K,J)=0.E0
      ERJZ(K,J)=0.E0
      ERJH(K,J)=0.E0
10    ERJF(K,J)=0.E0
      DO 5 MM=1,NM
      DISM(MM)=0.E0
5       XDM(MM)=0.E0
      RERZ=0.E0
      RERH=0.E0
      RERF=0.E0
      VALUES(5)=0.E0
      VALUES(6)=0.E0
C
C     NONDIMENSIONALIZE TRANSLATIONAL INITIAL CONDITIONS AND
C     ADJUST DIMENSIONS OF TRANSLATIONAL FORCING FUNCTIONS.
C
      IF(IANAL.EQ.0.AND.NFF.EQ.0.AND.IFOF.EQ.0) GO TO 17
      GO TO 19
```

```
17    IERR=1                                                        ASAN 590
      PRINT 18                                                      ASAN 600
18    FORMAT(1H ,94H*** NEITHER FORCING FUNCTIONS NOR INITIAL CONDITIONSASAN 610
     1 SPECIFIED FOR STATIC ANALYSIS. (ASAN) ***)                   ASAN 620
      RETURN                                                        ASAN 630
19    DO 20 J=1,NJ                                                  ASAN 640
      DO 20 K=1,2                                                   ASAN 650
      DIS(K,J)=DIS(K,J)/AVGL                                        ASAN 660
      VEL(K,J)=VEL(K,J)/AVGL                                        ASAN 670
      ACC(K,J)=ACC(K,J)/AVGL                                        ASAN 680
      BET(K,J)=BET(K,J)/AVGL                                        ASAN 690
      DAS(K,J)=DAS(K,J)*AVGL*AVGL                                   ASAN 700
20    FOR(K,J)=FOR(K,J)*AVGL                                        ASAN 710
      IF (NFF.EQ.0) GO TO 40                                        ASAN 720
      DO 30 I=1,NFF                                                 ASAN 730
      L=LFF+4*I                                                     ASAN 740
      K=KDATA(LFFI+5*I-2)                                           ASAN 750
      IF (K.EQ.3) GO TO 30                                          ASAN 760
      DATA(L-2)=DATA(L-2)*AVGL                                      ASAN 770
30    CONTINUE                                                      ASAN 780
C                                                                   ASAN 790
C     PERFORM STATIC ANALYSIS, IF NECESSARY, FOR INITIAL APPLIED FORCES.ASAN 800
C                                                                   ASAN 810
40    DTX=DT                                                        ASAN 820
      DT=0.E0                                                       ASAN 830
      IF ( IANAL.NE.0.AND.IFOR.EQ.0.AND.NFF.EQ.0) GO TO 125         ASAN 840
      DO 60 I=1,NDF                                                 ASAN 850
60    SOLN(I)=0.E0                                                  ASAN 860
      IF (IANAL.EQ.0) GO TO 75                                      ASAN 870
      IF ( IFOR.EQ.0 ) GO TO 125                                    ASAN 880
```

```
      CALL PAGE                                                         ASAN 890
50    FORMAT (1H ,25HSOLUTION FOR INITIAL TIME,0PE13.5//)               ASAN 900
      WRITE(NPRT,50) TIME                                               ASAN 910
      DO 70 J=1,NJ                                                      ASAN 920
      DO 70 K=1,3                                                       ASAN 930
      IF(ABS(FOR(K,J)/AVGL-TINY).LT.1.E2*TINY) FOR(K,J)=0.E0            ASAN 940
70    F(K,J)=FOR(K,J)                                                   ASAN 950
      GO TO 80                                                          ASAN 960
75    CALL PAGE                                                         ASAN 970
      WRITE(NPRT,77) TIME                                               ASAN 980
77    FORMAT(1H ,22HSOLUTION FOR LOAD STEP,0PE13.5//)                   ASAN 990
      IF(NFF.EQ.0) GO TO 82                                             ASAN1000
      CALL FORS (TIME)                                                  ASAN1010
82    DO 79 J=1,NJ                                                      ASAN1020
      DO 79 K=1,3                                                       ASAN1030
      IF(ABS(FOR(K,J)/AVGL-TINY).LT.1.E2*TINY) FOR(K,J)=0.E0            ASAN1040
79    F(K,J)=F(K,J)+FCR(K,J)                                            ASAN1050
C                                                                       ASAN1060
C     STATIC SOLUTION BY FUNCTIONAL MINIMIZATION.                       ASAN1070
C                                                                       ASAN1080
80    CALL SEEK (SOLN,VALUEM)                                           ASAN1090
      DO 90 I=1,NDFJ                                                    ASAN1100
      J=IDFI(I)                                                         ASAN1110
      K=IDFII(I)                                                        ASAN1120
90    XDJ(K,J)=SOLN(I)                                                  ASAN1130
      L=NDFJ+1                                                          ASAN1140
      DO 91 I=L,NDF                                                     ASAN1150
      M=IDFI(I)                                                         ASAN1160
91    XDM(M)=SOLN(I)                                                    ASAN1170
C                                                                       ASAN1180
```

```
C      CHECK STATUS OF ELASTIC-PLASTIC MEMBERS.                          ASAN1190
       IYFLAG = 0                                                        ASAN1200
       IF (IYLD.LT.NM) CALL TEST (IYFLAG)                                ASAN1210
C                                                                        ASAN1220
C      STORE ENERGIES AND STRESS RESULTANTS, AND CHECK MEMBER            ASAN1230
C      FAILURE CRITERIA.                                                 ASAN1240
       CALL FAIL                                                         ASAN1250
C                                                                        ASAN1260
C      REPEAT STATIC CALCULATION IF MEMBERS HAVE YIELDED.                ASAN1270
       IF (IYFLAG.EQ.1.AND.LERR.EQ.0.AND.IERR.EQ.0) GO TO 80             ASAN1280
C                                                                        ASAN1290
C      CALCULATE ERROR FOR STATIC ANALYSIS.                              ASAN1300
C                                                                        ASAN1310
       CALL ERRS (SOLN,VALUEM)                                           ASAN1320
       WRITE (NPRT,100) RERZ,RERH,RERF                                   ASAN1330
100    FORMAT (1H0,5X,39HTOTAL ERROR MEASURES,  ZERO-TIME ERROR=,0PE13.5/ASAN1340
      11H ,28X,16HHALF-TIME ERROR=,E13.5/1H ,28X,16HFULL-TIME ERROR=,E13.ASAN1350
      25//)                                                              ASAN1360
C                                                                        ASAN1370
C      SOLUTION PRINTOUT AND STRESS CALCULATIONS FOR INITIAL TIME POINT. ASAN1380
C                                                                        ASAN1390
       DO 120 J=1,NJ                                                     ASAN1400
       DO 120 K=1,3                                                      ASAN1410
120    DIS(K,J)=XDJ(K,J)                                                 ASAN1420
       DO 121 L=1,NM                                                     ASAN1430
121    DISM(L)=XDM(L)                                                    ASAN1440
       CALL OUTS                                                         ASAN1450
125    DT = DTX                                                          ASAN1460
       IF(IPLOT.EQ.1) CALL PLOG(NPLOT)                                   ASAN1470
       IF(IFAIL.EQ.1) RETURN                                             ASAN1480
```

```
      IF (IANAL.NE.0) RETURN                                            ASAN1490
      IF (NFF.EQ.0) GO TO 130                                           ASAN1500
      TIME = TIME + DT                                                  ASAN1510
      IF(ISTOP.EQ.0.AND.IREC.GT.0) GO TO 130                            ASAN1520
      CALL TICS(TIMN,IGA)                                               ASAN1530
      IF(IGA.EQ.1) GO TO 75                                             ASAN1540
  130 IF (ITAPE.NE.0) CALL REGO (-ITAPE)                                ASAN1550
      RETURN                                                            ASAN1560
      END                                                               ASAN1570
```

```
CBARS    0  10
      SUBROUTINE BARS (K,L,ADIM,DIAM,PERIM)                             BARS   0
C                                                                       BARS  10
C     THIS SUBROUTINE CALCULATES TOTAL AREA OF REBAR GROUPS, GIVEN BAR SBARS  20
C     AND NUMBER OF BARS IN GROUP                                       BARS  30
C                                                                       BARS  40
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), BARS  50
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      BARS  60
      INTEGER HEAD,DHEAD                                                BARS  70
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,BARS  80
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,BARS  90
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, BARS 100
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  BARS 110
     4    NTIMES,NVEL,IINITD                                            BARS 120
      DIMENSION DB(12)                                                  BARS 130
C                                                                       BARS 140
      DATA DB(1),DB(2),DB(3),DB(4) /.25E0,.375E0,.5E0,.625E0/           BARS 150
      DATA DB(5),DB(6),DB(7),DB(8)/.75E0,.875E0,1.E0,1.128E0/           BARS 160
      DATA DB(9),DB(10),DB(11),DB(12)/1.27E0,1.41E0,1.693E0,2.257E0/    BARS 170
C                                                                       BARS 180
      IF (IUNITS.GE.2) GO TO 30                                         BARS 190
      LM1=0                                                             BARS 200
      IF (L.GE.2.AND.L.LE.11) LM1=L-1                                   BARS 210
      IF (L.EQ.14) LM1=11                                               BARS 220
      IF (L.EQ.18) LM1=12                                               BARS 230
      IF (LM1.EQ.0) GO TO 20                                            BARS 240
      ANUM=K                                                            BARS 250
      ADIM=ANUM*PI*DB(LM1)*DB(LM1)/4.0E0                                BARS 260
      DIAM=DB(LM1)                                                      BARS 270
      PERIM=PI*DB(LM1)                                                  BARS 280
```

```
      L=LM1                                                             BARS 290
      GO TO 40                                                          BARS 300
10    FORMAT (67H ***INVALID BAR SIZE NUMBER.AREA OF REBAR GROUP CANNOT BARS 310
     1BE COMPUTED.,/,55H EFFECTIVE LENGTH OF REBAR CANNOT BE OBTAINED.(BBARS 320
     2ARS)***)                                                          BARS 330
20    WRITE (NPRT,10)                                                   BARS 340
      IERR=IERR+1                                                       BARS 350
      LINE=LINE+2                                                       BARS 360
      L=20                                                              BARS 370
      GO TO 40                                                          BARS 380
C                                                                       BARS 390
C     EUROPEAN (METRIC) BAR SIZES ARE INPUT IN FORM OF DIAMETER(MM)     BARS 400
C     AND IN INTEGER FORM(I.D. L=26 => BAR DIA. =26. MM. BAR SIZE       BARS 410
C     RUN FROM  5.0 TO 50.0 MM DIA. THIS SEGMENT CONVERTS BAR SIZES.    BARS 420
C     THE INPUT IN THIS SEGMENT ARE ASSUMED CORRECT.(I.D. IT IS ASSUMED BARS 430
C     THE SPECIFIED BAR SIZES EXIST.)                                   BARS 440
C                                                                       BARS 450
30    DIAM=L                                                            BARS 460
      DIAM=DIAM/1000.E0                                                 BARS 470
      PERIM=PI*DIAM                                                     BARS 480
      ANUM=K                                                            BARS 490
      ADIM=ANUM*PI*DIAM*DIAM/4.E0                                       BARS 500
      IF (LINE.GE.NL) CALL PAGE                                         BARS 510
      L=20                                                              BARS 520
40    RETURN                                                            BARS 530
      END                                                               BARS 540
```

```
CBEAM    0  10
      SUBROUTINE BEAM                                                   BEAM   0
C                                                                       BEAM  10
C     SUBROUTINE TO INPUT AND OUTPUT ELEMENT PROPERTIES                 BEAM  20
C                                                                       BEAM  30
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),       BEAM  40
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),BEAM  50
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)                      BEAM  60
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9), BEAM  70
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)BEAM  80
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), BEAM  90
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),      BEAM 100
     1    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)     BEAM 110
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), BEAM 120
     1    PI,PERF,PERH,PERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      BEAM 130
      INTEGER HEAD,DHEAD                                                BEAM 140
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,BEAM 150
     1    IRFC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,BEAM 160
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, BEAM 170
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  BEAM 180
     4    NTIMES,NVEL,IINITD                                            BEAM 190
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),BEAM 200
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),    BEAM 210
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),BEAM 220
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                BEAM 230
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45), BEAM 240
     5   YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)               BEAM 250
      DATA JD1,JD2/1HH,1HS/,JD3/1H1/,ID1,ID2/1HL,1HN/,ID3/4HNONE/       BEAM 260
C                                                                       BEAM 270
C     INITIALIZE MEMBER COUNTER AND REBAR GROUP COUNTERS                BEAM 280
```

```
C                                                                      BEAM 290
      IMEM=0                                                           BEAM 300
      ITIET=0                                                          BEAM 310
      IGRPT=0                                                          BEAM 320
      MAXM=0                                                           BEAM 330
      IM=0                                                             BEAM 340
      NCM=0                                                            BEAM 350
      NLS=0                                                            BEAM 360
      IBA=0                                                            BEAM 370
      NLSR=0                                                           BEAM 380
C                                                                      BEAM 390
C     INITIALIZE ARRAYS                                                BEAM 400
C                                                                      BEAM 410
      DO 130 MEM=1,NMD                                                 BEAM 420
      BMEM(MEM)=0.E0                                                   BEAM 430
      BPP(MEM)=0.E0                                                    BEAM 440
      D(MEM)=0.E0                                                      BEAM 450
      DP(MEM)=0.E0                                                     BEAM 460
      DPP(MEM)=0.E0                                                    BEAM 470
      DWF(MEM)=0.E0                                                    BEAM 480
      TFWF(MEM)=0.E0                                                   BEAM 490
      TWWF(MEM)=0.E0                                                   BEAM 500
      BWF(MEM)=0.E0                                                    BEAM 510
      XL(MEM)=0.E0                                                     BEAM 520
      EFLM(MEM)=0.E0                                                   BEAM 530
      HMEM(MEM)=0.E0                                                   BEAM 540
      HTOP(MEM)=0.E0                                                   BEAM 550
      HTWF(MEM)=0.E0                                                   BEAM 560
      XBEGM(MEM)=0.E0                                                  BEAM 570
      YLDS(MEM)=0.E0                                                   BEAM 580
```

```
      IP(MEM)=0                                                         BEAM 590
      IQ(MEM)=0                                                         BEAM 600
      MTIES(MEM)=0                                                      BEAM 610
      MSHEAR(MEM)=0                                                     BEAM 620
      MCODE(MEM)=0                                                      BEAM 630
      MATR(MEM)=0                                                       BEAM 640
      MATW(MEM)=0                                                       BEAM 650
      MSTAT(MEM)=0                                                      BEAM 660
      MTYPE(MEM)=0                                                      BEAM 670
      NTIES(MEM)=0                                                      BEAM 680
      NGRP(MEM)=0                                                       BEAM 690
      DO 120 I=1,6                                                      BEAM 700
      NSPAC(I,MEM)=0                                                    BEAM 710
      ATIES(I,MEM)=0.E0                                                 BEAM 720
      PDP(I,MEM)=0.E0                                                   BEAM 730
      PDF(I,MEM)=0.E0                                                   BEAM 740
      STIES(I,MEM)=0.E0                                                 BEAM 750
  120 XBEGS(I,MEM)=0.E0                                                 BEAM 760
      DO 130 IGRP=1,10                                                  BEAM 770
      AGRP(IGRP,MEM)=0.E0                                               BEAM 780
      EFFL(IGRP,MEM)=0.E0                                               BEAM 790
      XBEG(IGRP,MEM)=0.E0                                               BEAM 800
      YBAR(IGRP,MEM)=0.E0                                               BEAM 810
  130 MBAR(IGRP,MEM)=0                                                  BEAM 820
      DO 140 I=1,NLD                                                    BEAM 830
      IPL(I)=0                                                          BEAM 840
      IQL(I)=0                                                          BEAM 850
      DO 140 J=1,5                                                      BEAM 860
  140 SPRING(J,I)=0.E0                                                  BEAM 870
C                                                                       BEAM 880
```

```
C     READ A DATA CARD.                                                 BEAM 890
C                                                                       BEAM 900
150   FORMAT (20A4)                                                     BEAM 910
      READ (NCRD,150) DHEAD                                             BEAM 920
160   FORMAT (1H0,20A4,//)                                              BEAM 930
      CALL PAGE                                                         BEAM 940
      WRITE (NPRT,160) DHEAD                                            BEAM 950
      ICARD = 1                                                         BEAM 960
170   FORMAT (A4,1X,3I5,4A1,I1,A4,1X,5E10.0)                            BEAM 970
180   READ (NCRD,170) J1,I1,I2,I3,J2,J3,J4,J5,I4,J6,R1,R2,R3,R4,R5      BEAM 980
      ICARD=ICARD+1                                                     BEAM 990
C                                                                       BEAM1000
C     TEST FOR TYPE OF DATA BY INVESTIGATING J1.                        BEAM1010
      CALL FORK (J1,IBRNCH)                                             BEAM1020
C                                                                       BEAM1030
C     CHECK FOR MATERIAL NAME.                                          BEAM1040
      IF (IBRNCH.EQ.1) GO TO 200                                        BEAM1050
      IF (IBRNCH.EQ.9) GO TO 840                                        BEAM1060
      IF (IBRNCH.EQ.7.OR.IBRNCH.EQ.8) GO TO 200                         BEAM1070
      CALL MATY (J6,MATN)                                               BEAM1080
      IF (MATN.NE.0) GO TO 200                                          BEAM1090
190   FORMAT (1H ,12H***MATERIAL ,A4,10H FOR CARD ,A4,12H OF ELEMENT ,I3BEAM1100
     1,1H-,I3,48H IS NOT IN THE MATERIAL DATA BLOCK (BEAM).***    )     BEAM1110
      LINE=LINE+1                                                       BEAM1120
      IF (LINE.GT.NL) CALL PAGE                                         BEAM1130
      IACT=I2                                                           BEAM1140
      ISHEAR=I3                                                         BEAM1150
      IF (I1.EQ.0) IACT=IP(IMEM)                                        BEAM1160
      IF (I1.EQ.0) ISHEAR=IQ(IMEM)                                      BEAM1170
      PRINT 190, J6,J1,IACT,ISHEAR                                      BEAM1180
```

```
      IF (I2.EQ.0.AND.I3.EQ.0) IACT=IP(IMEM)                            BEAM1190
      IF (I2.EQ.0.AND.I3.EQ.0) ISHEAR=IQ(IMEM)                          BEAM1200
      IERR=IERR+1                                                       BEAM1210
200   GO TO (210,340,400,530,530,680,740,180), IBRNCH                   BEAM1220
C                                                                       BEAM1230
C     INTERPRET ELEMENT PARAMETER CARD.                                 BEAM1240
C                                                                       BEAM1250
210   IZZ=1                                                             BEAM1260
220   IGRPT=0                                                           BEAM1270
      ITIET=0                                                           BEAM1280
      IMEM=IMEM+1                                                       BEAM1290
      IM=IMEM                                                           BEAM1300
      IF (IMEM.NE.I1.AND.I1.GT.0) IM=I1                                 BEAM1310
      NDF=NDF+1                                                         BEAM1320
      IF(NDF.LE.NDFD)GO TO 225                                          BEAM1330
      IERR=IERR+1                                                       BEAM1340
      PRINT 226,NDFD                                                    BEAM1350
226   FORMAT (1H ,37H ***NO. OF DEGREES OF FREEDOM EXCEEDS,I5,65H,LARGESBEAM1360
     1T NO. OF DEGREES OF FREEDOM ALLOWED BY PROGRAM (BEAM). ***)       BEAM1370
225   IDFI(NDF)=IM                                                      BEAM1380
C                                                                       BEAM1390
C     CHECK FOR VALID NODAL POINT INFORMATION                           BEAM1400
      ISK=0                                                             BEAM1410
      IF (I2.LE.NJD.AND.I3.LE.NJD) GO TO 240                            BEAM1420
      IF (LINE.GT.NL) CALL PAGE                                         BEAM1430
      ISK=1                                                             BEAM1440
      IERR=IERR+1                                                       BEAM1450
      LINE=LINE+1                                                       BEAM1460
230   FORMAT (1H ,11H***ELEMENT ,I3,1H-,I3,56H HAS A JOINT NUMBER WHICH BEAM1470
     1EXCEEDS ARRAY SIZE (BEAM).*** )                                   BEAM1480
```

```
      PRINT 230, I2,I3
240   IF (I2.LE.NJ.AND.I3.LE.NJ) GO TO 260
      IF (LINE.GT.NL) CALL PAGE
      IERR=IERR+1
      LINE=LINE+1
      ISK=1
250   FORMAT (1H ,11H***ELEMENT ,I3,1H-,I3,56H HAS A JOINT WHICH IS NOT
     1IN JOINT DATA BLOCK (BEAM).***)
      PRINT 250, I2,I3
260   IF (I2.NE.I3) GO TO 280
      IF (LINE.GT.NL) CALL PAGE
      IERR=IERR+1
      LINE=LINE+1
      ISK=1
270   FORMAT (1H ,11H***ELEMENT ,I3,1H-,I3,46H BEGINS AND ENDS AT THE SA
     1ME JOINT (BEAM).*** )
      PRINT 270, I2,I3
C
C     CHECK FOR VALID ELEMENT NUMBER.
280   IF (IM.LE.NMD) GO TO 300
      IF (LINE.GT.NL) CALL PAGE
      IERR=IERR+1
      LINE=LINE+1
290   FORMAT (1H ,11H***ELEMENT ,I3,1H-,I3,42H REQUIRES EXCEEDING ARRAY
     1SIZE (BEAM).*** )
      PRINT 290, I2,I3
C
C     IF NO NODAL POINT ERRORS, CALCULATE GROSS AND EFFECTIVE LENGTHS.
C
300   IF (ISK.EQ.1) GO TO 330
```

AD-A038 317 VIRGINIA POLYTECHNIC INST AND STATE UNIV BLACKSBURG --ETC F/G 13/13
RELIABILITY STUDY OF SINGER. VOLUME II. USER'S MANUAL.(U)
JAN 77 A E SOMERS, S M HOLZER, J C BRADSHAW F29601-75-C-0050

UNCLASSIFIED AFWL-TR-76-192-VOL-2 NL

2 OF 4
AD
A038317

2 OF 4

38317

1.0 4.5 5.0 5.6 6.3 7.1 8.0 2.8 3.2 3.6 4.0 2.5 2.2 2.0

1.1 1.8

1.25 1.4 1.6

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
      GROSL=SQRT((X(I2)-X(I3))**2+(Y(I2)-Y(I3))**2)
      XL(IM)=GROSL
      EFLM(IM) = XL(IM)
      IF (IZZ.EQ. 2) GO TO 330
      IF ((R4+R5).LE.GROSL) GO TO 330
      IF (LINE.GT.NL) CALL PAGE
      IERR=IERR+1
      LINE=LINE+1
320   FORMAT (1H ,11H***ELEMENT ,I3,1H-,I3,52H HAS LONGER JOINT SPANS TH
     1AN ITS LENGTH (BEAM).***  )
      PRINT 320, I2,I3
330   IF (IM.GT.MAXM) MAXM=IM
C
C     INTERPRET ALPHAMERIC DATA
C
      IACT=2
      IF (J2.EQ.ID1) IACT=1
      IF (J2.EQ.ID2) IACT=3
      ISHEAR=0
      IF (J3.EQ.JD2) ISHEAR=1
      IBOND=0
      IF (J4.EQ.JD1) IBOND=1
C
C     ASSIGN ELEMENT PARAMETERS TO ARRAYS
      IF (IM.GT.NMD) GO TO 180
      IP(IM)=I2
      IQ(IM)=I3
      MSTAT(IM)=IACT
      MSHEAR(IM)=ISHEAR
      IF (IZZ.EQ.2) GO TO 720
```

```
      HTOP(IM) = R1                                                     BEAM2090
      D(IM)=R2                                                          BEAM2100
      DP(IM)=R3                                                         BEAM2110
      XBEGM(IM)=R4                                                      BEAM2120
      EFLM(IM)=XL(IM)-R4-R5                                             BEAM2130
      GO TO 180                                                         BEAM2140
C                                                                       BEAM2150
C     INTERPRET CONCRETE PARAMETER CARD                                 BEAM2160
C                                                                       BEAM2170
340   IM2=IM                                                            BEAM2180
      IF (IM.NE.I1.AND.I1.NE.0) IM2=I1                                  BEAM2190
C                                                                       BEAM2200
C     CHECK FOR ARRAY SPACE AVAILABILITY                                BEAM2210
      IF (IM2.LE.NMD) GO TO 360                                         BEAM2220
      IERR=IERR+1                                                       BEAM2230
      LINE=LINE+1                                                       BEAM2240
      IF (LINE.GT.NL) CALL PAGE                                         BEAM2250
350   FORMAT (1H ,11H***ELEMENT ,I3,1H-,I3,44H REQUIRES EXCEEDING ARRAY BEAM2260
     1SIZE (BEAM).***    )                                              BEAM2270
      PRINT 350, IP(IM2),IQ(IM2)                                        BEAM2280
C                                                                       BEAM2290
C     COMPARE CONFINED AND OVERALL DIMENSIONS.                          BEAM2300
360   IF (R1.GE.R3.AND.R2.GE.R4) GO TO 380                              BEAM2310
      IREC=IREC+1                                                       BEAM2320
      LINE=LINE+1                                                       BEAM2330
      IF (LINE.GT.NL) CALL PAGE                                         BEAM2340
370   FORMAT (1H ,29H***CAGE DIMENSIONS OF ELEMENT,I3,1H-,I3,50H EXCEED BEAM2350
     1GROSS CROSS SECTION DIMENSIONS (BEAM).*** )                       BEAM2360
      PRINT 370, IP(IM2),IQ(IM2)                                        BEAM2370
C                                                                       BEAM2380
```

```
C       ASSIGN DATA TO ARRAYS.                                          BEAM2390
380     CALL MATY (J1,M1)                                               BEAM2400
        IF (M1.NE.0) GO TO 390                                          BEAM2410
        LINE=LINE+1                                                     BEAM2420
        IF (LINE.GT.NL) CALL PAGE                                       BEAM2430
        PRINT 190, J1,J1,IP(IM2),IQ(IM2)                                BEAM2440
390     MATR(IM2)=MATN                                                  BEAM2450
        MCODE(IM2)=M1                                                   BEAM2460
        HMEM(IM2)=R1                                                    BEAM2470
        BMEM(IM2)=R2                                                    BEAM2480
        DPP(IM2)=R3                                                     BEAM2490
        BPP(IM2)=R4                                                     BEAM2500
        GO TO 180                                                       BEAM2510
C                                                                       BEAM2520
C       INTERPRET LONGITUDINAL REBAR DATA.                              BEAM2530
C                                                                       BEAM2540
400     IGRPT=IGRPT+1                                                   BEAM2550
        IMR=IM                                                          BEAM2560
        IGRP=IGRPT                                                      BEAM2570
        IF (IGRPT.NE.I1.AND.I1.NE.0) IGRP=I1                            BEAM2580
C                                                                       BEAM2590
C       CHECK ARRAY SPACE AVAILABILITY                                  BEAM2600
        IF (IMR.LE.NMD) GO TO 410                                       BEAM2610
        LINE=LINE+1                                                     BEAM2620
        IF (LINE.GT.NL) CALL PAGE                                       BEAM2630
        PRINT 350, IP(IM),IQ(IM)                                        BEAM2640
        IERR=IERR+1                                                     BEAM2650
410     IF (IGRP.LE.10) GO TO 430                                       BEAM2660
        LINE=LINE+1                                                     BEAM2670
        IF (LINE.GT.NL) CALL PAGE                                       BEAM2680
```

```
420   FORMAT (1H ,51H***THE NUMBER OF LONGITUDINAL REINFORCEMENT GROUPS,BEAM2690
     1,I3,12H OF ELEMENT ,I3,1H-,I3,40HREQUIRES EXCEEDING ARRAY SIZE (BEBEAM2700
     2AM).***)                                                          BEAM2710
      PRINT 420, IGRP,IP(IM),IQ(IM)                                     BEAM2720
      ISK=1                                                             BEAM2730
      IERR=IERR+1                                                       BEAM2740
C                                                                       BEAM2750
C     CALCULATE MATERIAL NUMBER FOR LONGITUDINAL REBAR                  BEAM2760
430   IF (IMR.GT.NMD.OR.IGRP.GT.10) GO TO 440                           BEAM2770
      IF (IGRP.GT.NGRP(IMR)) NGRP(IMR)=IGRP                             BEAM2780
440   M=0                                                               BEAM2790
      N=0                                                               BEAM2800
      IF (J2.EQ.JD3) M=1                                                BEAM2810
      IF (J3.EQ.JD3) N=1                                                BEAM2820
C                                                                       BEAM2830
C     CALCULATE AND CHECK EFFECTIVE LENGTH OF REBAR.                    BEAM2840
      BRLEN=XL(IMR)-R3-R4                                               BEAM2850
      IF (BRLEN.GE.0.) GO TO 460                                        BEAM2860
      IF (ISK.EQ.1) GO TO 460                                           BEAM2870
450   FORMAT (1H ,27H***LONGITUDINAL REBAR GROUP,I3,11H OF ELEMENT,I3,1HBEAM2880
     1-,I3,40H IS LONGER THAN THE ELEMENT (BEAM).***  )                 BEAM2890
      IF (LINE.GT.NL) CALL PAGE                                         BEAM2900
      IERR=IERR+1                                                       BEAM2910
      LINE=LINE+1                                                       BEAM2920
      PRINT 450, IGRP,IP(IM),IQ(IM)                                     BEAM2930
460   IF(R1.GT.0.E0) GO TO 490                                          BEAM2940
      IF (I3.LE.1) GO TO 490                                            BEAM2950
      CALL BARS (I2,I3,R1,DIAM,PERIM)                                   BEAM2960
      IF (ISK.EQ.1) GO TO 500                                           BEAM2970
      IF (IBOND.EQ.0) GO TO 490                                         BEAM2980
```

```
      CALL BOND (IGRP,IMR,R2,M,N,DIAM,R3,I3,BRLEN,IN)                   BEAM2990
      IF (IN.NE.0) GO TO 480                                            BEAM3000
      IF (IBA.EQ.IMR) GO TO 480                                         BEAM3010
      LINE=LINE+1                                                       BEAM3020
      IF (LINE.GT.NL) CALL PAGE                                         BEAM3030
470   FORMAT (1H ,44H ACI 318-71 BOND CHECK WAS MADE FOR ELEMENT ,I3,1H-BEAM3040
     1,I3,1H.)                                                          BEAM3050
      WRITE (NPRT,470) IP(IMR),IQ(IMR)                                  BEAM3060
      IBA=IMR                                                           BEAM3070
480   IF (I3.EQ.20.OR.IBOND.EQ.0) GO TO 490                             BEAM3080
      CALL ENDS(IMR,M,N,R3,R2,BRLEN,DIAM)                               BEAM3090
C                                                                       BEAM3100
C     ASSIGN LONGITUDINAL REBAR INFORMATION TO ARRAYS.                  BEAM3110
490   IF (IGRP.GT.10.OR.IMR.GT.NMD) GO TO 500                           BEAM3120
      AGRP(IGRP,IMR)=R1                                                 BEAM3130
      EFFL(IGRP,IMR)=BRLEN                                              BEAM3140
      YBAR(IGRP,IMR)=R2                                                 BEAM3150
      MBAR(IGRP,IMR)=MATN                                               BEAM3160
      XBEG(IGRP,IMR)=R3                                                 BEAM3170
      BDM(IGRP,IMR)=DIAM                                                BEAM3180
C     CHECK FOR A NEGATIVE EFFECTIVE LENGTH OF REBAR                    BEAM3190
      IF(BRLEN.GT.0.E0) GO TO 500                                       BEAM3200
      IERR = IERR + 1                                                   BEAM3210
      LINE = LINE + 1                                                   BEAM3220
      PRINT 495, IGRP,IP(IMR),IQ(IMR),BRLEN                             BEAM3230
495   FORMAT(1H ,28H*** LONGITUDINAL REBAR GROUP ,I3,11H OF ELEMENT,I3, BEAM3240
     1 2H -,I3,28H HAS AN EFFECTIVE LENGTH OF ,G14.7,4H *** )           BEAM3250
C                                                                       BEAM3260
C     CHECK VALIDITY OF LONGITUDINAL REBAR GROUP AREA.                  BEAM3270
500   IF (R1.GT.0.E0) GO TO 180                                         BEAM3280
```

```
      IF (LINE.GT.NL) CALL PAGE                                         BEAM3290
  510 FORMAT (28H ***LONGITUDINAL REBAR GROUP,I3,12H OF ELEMENT  ,I3,1H-,BEAM3300
     1I3,45H HAS A ZERO CROSS SECTIONAL AREA (BEAM).***  )              BEAM3310
      PRINT 510, IGRP,IP(IM),IQ(IM)                                     BEAM3320
      IERR=IERR+1                                                       BEAM3330
      LINE=LINE+1                                                       BEAM3340
      IF (R4.GT.0.E0) GO TO 180                                         BEAM3350
      R4=-R4                                                            BEAM3360
  520 FORMAT (1H ,27H*THE LENGTH OF REBAR GROUP ,I3,12H OF ELEMENT ,I3,1BEAM3370
     1H-,I3,42H HAS BEEN REVISED TO BE POSITIVE (BEAM).* )              BEAM3380
      PRINT 520, IGRP,IP(IM),IQ(IM)                                     BEAM3390
      IREC=IREC+1                                                       BEAM3400
      GO TO 180                                                         BEAM3410
C                                                                       BEAM3420
C     INTERPRET LATERAL REBAR DATA                                      BEAM3430
C                                                                       BEAM3440
C     ASSIGN MEMBER AND GROUP DIMENSIONS                                BEAM3450
  530 ITIET=ITIET+1                                                     BEAM3460
      ISK=0                                                             BEAM3470
      ITI=ITIET                                                         BEAM3480
      PDPT=0.E0                                                         BEAM3490
      IF (ITI.NE.I1.AND.I1.NE.0) ITI=I1                                 BEAM3500
      IMT=IM                                                            BEAM3510
C                                                                       BEAM3520
C     CHECK AVAILABILITY OF ARRAY SPACE.                                BEAM3530
      IF (IMT.LE.NMD) GO TO 550                                         BEAM3540
      IF (LINE.GT.NL) CALL PAGE                                         BEAM3550
      ISK=1                                                             BEAM3560
  540 FORMAT (1H ,48H***LATERAL REINFORCEMENT IS ASSIGNED TO ELEMENT  ,I3BEAM3570
     1,1H-,I3,55H. THIS ELEMENT REQUIRES EXCEEDING ARRAY SIZE (BEAM).***BEAM3580
```

```
      2)                                                                BEAM3590
      PRINT 540, IP(IM),IQ(IM)                                          BEAM3600
      IERR=IERR+1                                                       BEAM3610
      LINE=LINE+1                                                       BEAM3620
  550 IF (ITI.GT.0.AND.ITI.LE.6) GO TO 570                              BEAM3630
      IF (LINE.GT.NL) CALL PAGE                                         BEAM3640
      ISK=1                                                             BEAM3650
  560 FORMAT (1H ,11H***ELEMENT ,I3,1H-,I3,5H HAS ,I3,50H TIE OR STIRRUPBEAM3660
     1 GROUPS. SIX ARE ALLOWED (BEAM).***)                              BEAM3670
      PRINT 560, IP(IM),IQ(IM),ITI                                      BEAM3680
      IERR=IERR+1                                                       BEAM3690
      LINE=LINE+2                                                       BEAM3700
C                                                                       BEAM3710
C     CALCULATE LATERAL GROUP REBAR AREA AND CHECK VALIDITY OF AREA     BEAM3720
  570 IF (R1.GT.0.E0) GO TO 600                                         BEAM3730
      IF (I2.NE.0.AND.I4:NE.0) GO TO 590                                BEAM3740
      IF (LINE.GT.NL) CALL PAGE                                         BEAM3750
  580 FORMAT (1H ,11H***ELEMENT ,I3,1H-,I3,62H HAS A LATERAL REINFORCEMEBEAM3760
     1NT GROUP WITH ZERO AREA (BEAM).***  )                             BEAM3770
      PRINT 580, IP(IM),IQ(IM)                                          BEAM3780
      IERR=IERR+1                                                       BEAM3790
      LINE=LINE+1                                                       BEAM3800
  590 CALL BARS (I4,I2,R1,DIAM,PERIM)                                   BEAM3810
C                                                                       BEAM3820
C     ASSIGN LATERAL GROUP REBAR INFORMATION TO ARRAYS.                 BEAM3830
C     CALCULATE MATERIAL NUMBER.                                        BEAM3840
  600 IF(R4.NE.0.E0.OR.(I2.NE.0.AND.I4.NE.0)) GO TO 620                 BEAM3850
      IF (LINE.GT.NL) CALL PAGE                                         BEAM3860
  610 FORMAT (74H ***INSUFFICIENT INFOPMATION PROVIDED TO DETERMINE STIRBEAM3870
     1RUP VOLUME OF GROUP,I3,11H IN ELEMENT,I3,1H-,I3,/,67H WHICH IS NEEBEAM3880
```

```
      2DED IN COMPUTING THE CONFINEMENT COEFFICIENT.(BEAM)***)
       PRINT 610, ITI,IP(IM),IQ(IM)
       IERR=IERR+1
       ISK=1
       LINE=LINE+1
C
C      CALCULATE CONFINEMENT FACTOR.
620    IF (R2.GT.0.E0) GO TO 640
630    FORMAT (23H ***TIE (STIRRUP) GROUP,I2,11H OF ELEMENT,I3,1H-,I3,39H
      1 HAS ZERO OR NEGATIVE SPACING.(BEAM)***)
       IF (LINE.GT.NL) CALL PAGE
       LINE=LINE+1
       IERR=IERR+1
       PRINT 630, ITI,IP(IM),IQ(IM)
       GO TO 650
640    IF (ISK.EQ.1) GO TO 650
       VSTIR=R4
       IF (R4.EQ.0.E0) VSTIR=R1*DPP(IMT)+BPP(IMT)*PI*DIAM*DIAM/2.E0
       ZZA=R2*BPP(IMT)*DPP(IMT)
       IF (ZZA.EQ.0.E0) GO TO 650
       PPP=VSTIR/ZZA
       PDPT=PPP*SQRT(BPP(IMT)/R2)
C
C      ASSIGN DATA TO ARRAYS.
650    IF (I3.NE.0) GO TO 670
       LINE=LINE+1
       IF (LINE.GT.NL) CALL PAGE
660    FORMAT (36H**NUMBER OF TIES (STIRRUPS) IN GROUP,I2,11H OF ELEMENT,
      1I3,1H-,I3,17H IS ZERO.(BEAM)**)
       PRINT 660, ITI,IP(IM),IQ(IM)
```

```
      IREC=IREC+1                                                       BEAM4190
670   IF (IMT.GT.NMD.OR.ITI.GT.6) GO TO 180                             BEAM4200
      ATIES(ITI,IMT)=R1                                                 BEAM4210
      STIES(ITI,IMT)=R2                                                 BEAM4220
      XBEGS(ITI,IMT)=R3                                                 BEAM4230
      MTYPE(IMT)=IBRNCH-3                                               BEAM4240
      POP(ITI,IMT)=PDPT                                                 BEAM4250
      PDF(ITI,IMT)=PPP*FCFY(MATN)                                       BEAM4260
      NSPAC(ITI,IMT)=I3                                                 BEAM4270
      MTIES(IMT)=MATN                                                   BEAM4280
      YLDS(IMT)=FCFY(MATN)                                              BEAM4290
      IF (ITI.GT.NTIES(IMT)) NTIES(IMT)=ITI                             BEAM4300
      IF (MTYPE(IMT).EQ.2) NSPAC(6,IMT)=-1                              BEAM4310
      GO TO 180                                                         BEAM4320
C                                                                       BEAM4330
C     INTERPRET AND STORE DATA ON WIDE FLANGE TYPE BEAMS.               BEAM4340
C                                                                       BEAM4350
680   IMW=IM                                                            BEAM4360
      IF (IMW.NE.I1.AND.I1.NE.0) IMW=I1                                 BEAM4370
      IF (IMW.LE.NMD) GO TO 690                                         BEAM4380
      LINE=LINE+1                                                       BEAM4390
      IF (LINE.GT.NL) CALL PAGE                                         BEAM4400
      PRINT 290, IP(IMW),IQ(IMW)                                        BEAM4410
      IERR=IERR+1                                                       BEAM4420
C                                                                       BEAM4430
C     CHECK FOR VALIDITY OF DIMENSIONS                                  BEAM4440
690   IF (R5.GT.0.E0.AND.R2.GT.0.E0.AND.R3.GT.0.E0.AND.R4.GT.0.E0) GO TOBEAM4450
     1 710                                                              BEAM4460
      LINE=LINE+1                                                       BEAM4470
      IF (LINE.GT.NL) CALL PAGE                                         BEAM4480
```

```
700   FORMAT (49H **A DIMENSION OF THE STEEL BEAM IS ZERO.(BEAM)**)       BEAM4490
      PRINT 700                                                           BEAM4500
      IREC=IREC+1                                                         BEAM4510
710   IF (IMW.GT.NMD) GO TO 730                                           BEAM4520
      MTYPE(IMW)=3                                                        BEAM4530
      NCM=NCM+1                                                           BEAM4540
      IF (I2.EQ.0.AND.I3.EQ.0) GO TO 720                                  BEAM4550
      MTYPE(IMW)=4                                                        BEAM4560
      NAME(9)=ID3                                                         BEAM4570
      IZZ=2                                                               BEAM4580
      GO TO 220                                                           BEAM4590
720   CONTINUE                                                            BEAM4600
      DWF(IMW)=R2                                                         BEAM4610
      TFWF(IMW)=R5                                                        BEAM4620
      TWWF(IMW)=R3                                                        BEAM4630
      BWF(IMW)=R4                                                         BEAM4640
      HTWF(IMW)=R1                                                        BEAM4650
730   IF (IMW.LE.NMD) MATW(IMW)=MATN                                      BEAM4660
      GO TO 180                                                           BEAM4670
C                                                                         BEAM4680
C     INTERPRET LEAF SPRING DATA CARD.                                    BEAM4690
C                                                                         BEAM4700
740   NLS=NLS+1                                                           BEAM4710
      ILS=NLS                                                             BEAM4720
      ISK=0                                                               BEAM4730
      IF (ILS.NE.I1.AND.I1.NE.0) ILS=I1                                   BEAM4740
      IF (ILS.LE.NLD) GO TO 760                                           BEAM4750
      IF (LINE.GT.NL) CALL PAGE                                           BEAM4760
      LINE=LINE+1                                                         BEAM4770
      IERR=IERR+1                                                         BEAM4780
```

```
750   FORMAT (1H ,23H***LEAF SPRING ELEMENT ,I3,1H-,I3,52H REQUIRES EXCEBEAM4790
     1EDING +SPRING+ ARRAY SIZE (BEAM).***  )                            BEAM4800
      PRINT 750, I2,I3                                                  BEAM4810
      ISK=1                                                             BEAM4820
760   IF (I2.NE.0.AND.I3.NE.0) GO TO 780                                BEAM4830
      IF (LINE.GT.NL) CALL PAGE                                         BEAM4840
770   FORMAT (15H ***LEAF SPRING,I3,1H-,I3,38H HAS AN UNSPECIFIED JOINT BEAM4850
     1(BEAM).***  )                                                     BEAM4860
      PRINT 770, I2,I3                                                  BEAM4870
      LINE=LINE+1                                                       BEAM4880
      IERR=IERR+1                                                       BEAM4890
      ISK=1                                                             BEAM4900
780   IF (I2.LE.NJ.AND.I3.LE.NJ) GO TO 800                              BEAM4910
      IF (LINE.GT.NL) CALL PAGE                                         BEAM4920
      LINE=LINE+1                                                       BEAM4930
      IERR=IERR+1                                                       BEAM4940
790   FORMAT (15H ***LEAF SPRING,I3,1H-,I3,65H HAS A NODE WHICH WAS NOT BEAM4950
     1DESCRIBED BY NODAL POINT DATA.(BEAM)***)                          BEAM4960
      PRINT 790, I2,I3                                                  BEAM4970
800   IF (R1.LT.0.E0.OR.R3.LT.0.E0.OR.R4.LT.0.E0) GO TO 810             BEAM4980
      DET=R1*R3-R2*R2                                                   BEAM4990
      IF (DET.GE.0.E0) GO TO 830                                        BEAM5000
810   IREC=IREC+1                                                       BEAM5010
      IF (LINE.GT.NL) CALL PAGE                                         BEAM5020
820   FORMAT (1H ,23H***LEAF SPRING ELEMENT ,I3,1H-,I3,79H HAS AN UNACCEBEAM5030
     1PTABLE FLEXIBILITY MATRIX. IT IS NOT POSITIVE DEFINITE (BEAM).***)BEAM5040
      IERR=IERR+1                                                       BEAM5050
      PRINT 820, I2,I3                                                  BEAM5060
      LINE=LINE+1                                                       BEAM5070
830   IF (ISK.EQ.1) GO TO 180                                           BEAM5080
```

```
      IF (R1.EQ.0.E0) NLSR=NLSR+1                                       BEAM5090
      IF (R3.EQ.0.E0) NLSR=NLSR+1                                       BEAM5100
      IF (R4.EQ.0.E0) NLSR=NLSR+1                                       BEAM5110
      IF (R4.EQ.0.E0) R4=TINY*1.E04                                     BEAM5120
      IPL(ILS)=I2                                                       BEAM5130
      IQL(ILS)=I3                                                       BEAM5140
C     INVERT FLEXIBILITY MATRIX TO OBTAIN STIFFNESS MATRIX.             BEAM5150
      SPRING(4,ILS)=1.E0/R4                                             BEAM5160
      SPRING(5,ILS)=R5                                                  BEAM5170
      IF(DET.GT.0.E0) GO TO 834                                         BEAM5180
      SPRING(1,ILS)=1.E-04/TINY                                         BEAM5190
      SPRING(2,ILS)=0.E0                                                BEAM5200
      SPRING(3,ILS)=1.E-04/TINY                                         BEAM5210
      GO TO 180                                                         BEAM5220
834   SPRING(1,ILS)=R3/DET                                              BEAM5230
      SPRING(2,ILS)=-R2/DET                                             BEAM5240
      SPRING(3,ILS)=R1/DET                                              BEAM5250
      GO TO 180                                                         BEAM5260
C                                                                       BEAM5270
C     IF END OF DATA, SET NM EQUAL TO THE MAXIMUM COUNTER VALUE         BEAM5280
C                                                                       BEAM5290
840   NM=IMEM                                                           BEAM5300
C                                                                       BEAM5310
C     CONDUCT ADDITIONAL CHECKS ON NATURE OF INPUT DATA.                BEAM5320
      DO 880 I=1,NM                                                     BEAM5330
      NGP=NGRP(I)                                                       BEAM5340
      IF (NGP.EQ.0) GO TO 880                                           BEAM5350
      DO 870 J=1,NGP                                                    BEAM5360
      IF (YBAR(J,I).GT.HTOP(I)) GO TO 850                               BEAM5370
      IF (YBAR(J,I).LT.(HTOP(I)-HMEM(I))) GO TO 850                     BEAM5380
```

```
      GO TO 870                                                         BEAM5390
  850 IF (LINE.GT.NL) CALL PAGE                                         BEAM5400
  860 FORMAT (23H ***REINFORCEMENT GROUP,I3,12H OF ELEMENT ,I3,1H-,I3,68BEAM5410
     1H IS NOT CONTAINED IN THE CONCRETE PORTION OF THE ELEMENT (BEAM).*BEAM5420
     2** )                                                              BEAM5430
      PRINT 860, J,IP(I),IQ(I)                                          BEAM5440
      IERR=IERR+1                                                       BEAM5450
      LINE=LINE+1                                                       BEAM5460
  870 CONTINUE                                                          BEAM5470
  880 CONTINUE                                                          BEAM5480
      IF (MAXM.EQ.NM) GO TO 900                                         BEAM5490
  890 FORMAT (62H ***ELEMENTS WERE NOT INPUT WITH CONSECUTIVE NUMBERS.(BBEAM5500
     1EAM)***)                                                          BEAM5510
      PRINT 890                                                         BEAM5520
      IERR=IERR+1                                                       BEAM5530
      LINE=LINE+1                                                       BEAM5540
  900 DO 920 I=1,NM                                                     BEAM5550
      IF (EFLM(I).EQ.0.E0) GO TO 920                                    BEAM5560
      RATIO=HMEM(I)/EFLM(I)                                             BEAM5570
      IF (RATIO.LT..4) GO TO 920                                        BEAM5580
      IF (LINE.GT.NL) CALL PAGE                                         BEAM5590
  910 FORMAT (48H **USING CRITERION OF 10.7, ACI 318-71, ELEMENT ,I3,1H-BEAM5600
     1,I3,24H IS A DEEP BEAM(BEAM).**)                                  BEAM5610
      PRINT 910, IP(I),IQ(I)                                            BEAM5620
      LINE=LINE+1                                                       BEAM5630
  920 CONTINUE                                                          BEAM5640
      RETURN                                                            BEAM5650
      END                                                               BEAM5660
```

```
CBODY    0  10
      SUBROUTINE BODY
C
C     THIS SUBROUTINE CALCULATES BODY FORCES AND MEMBER MASSES
C     FROM MEMBER WEIGHTS AND LUMPS THEM AT THE JOINTS.
C          L=0, DENSITY OF MEMEBER EQUALS ZERO.
C          L=1, DATA INCLUDES CALCULATED MEMBER MASSES.
C
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),
     1   MATH(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),
     2   MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9),
     1   SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50),
     1   ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),
     1   XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20),
     1   PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB
      INTEGER HEAD,DHEAD
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,
     1   IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,
     2   NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,
     3   NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,
     4   NTIMES,NVEL,IINITO
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),
     1   BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),
     2   HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),
     3   TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),
     4   XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45),
     5  YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)
```

```
      COMMON/SCALE/EGSIF,EGSIL
C
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     BEGIN DO LOOP TO CALCULATE WEIGHT AND MASS OF ALL MEMBERS.
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IFOR = 0
      L=0
      DO 10 I=1,NJ
      DO 10 J=1,3
10    FOR(J,I)=TINY
      IF (NM.EQ.0) GO TO 40
      DO 30 M=1,NM
      IPM=IP(M)
      IQM=IQ(M)
      IF(MTYPE(M).LE.3) GO TO 20
      MT=MATW(M)
      AREA=2.E0*TFWF(M)*BWF(M)+TWWF(M)*(DWF(M)-2.E0*TFWF(M))
      GO TO 25
20    MT=MCODE(M)
      AREA=BMEM(M)*HMEM(M)
25    IF(DENS(MT).EQ.0.E0) GO TO 30
      IFOR=1
      NMAS = NMAS + 2
C
C     COMPUTE WEIGHT IN UNITS OF INPUT DATA.
      WT = 0.5E0*AREA*XL(M)*DENS(MT)/EGSIF
C
C     STORE LUMPED BODY FORCE AT MEMBER JOINTS.
C     GRAVITY ASSUMED IN NEGATIVE Y GLOBAL DIRECTION.
```

```
C                                                                       BODY 590
      FOR(2,IPM) = FOR(2,IPM) - WT*EGSIF                                BODY 600
      FOR(2,IQM) = FOR(2,IQM) - WT*EGSIF                                BODY 610
      IF(IANAL.EQ.0) GO TO 30                                           BODY 620
      L=1                                                               BODY 630
C                                                                       BODY 640
C     STORE LUMPED MASSES AT MEMBER JOINTS.                             BODY 650
C                                                                       BODY 660
C    COMPUTE MASS IN UNITS OF INPUT DATA.                               BODY 670
      IF (IPRINT.LE.1) WT = WT/386.088E0                                BODY 680
      IF (IPRINT.GE.2) WT = WT/9.80665E0                                BODY 690
      AMOM = WT*XL(M)*XL(M)/12.E0/EGSIL**2                              BODY 700
C                                                                       BODY 710
C     ACCUMULATE MASS COMPONENTS IN UNITS OF INPUT DATA.                BODY 720
      DAS(1,IPM)=DAS(1,IPM)+WT                                          BODY 730
      DAS(2,IPM)=DAS(2,IPM)+WT                                          BODY 740
      DAS(3,IPM)=DAS(3,IPM)+AMOM                                        BODY 750
      DAS(1,IQM)=DAS(1,IQM)+WT                                          BODY 760
      DAS(2,IQM)=DAS(2,IQM)+WT                                          BODY 770
      DAS(3,IQM)=DAS(3,IQM)+AMOM                                        BODY 780
C                                                                       BODY 790
C     END OF DO LOOP                                                    BODY 800
30    CONTINUE                                                          BODY 810
C                                                                       BODY 820
C     CALL PRINTOUT ROUTINE, IF REQUIRED.                               BODY 830
C                                                                       BODY 840
40    IF (NMAS.NE.0.OR.L.NE.0) CALL LUMP                                BODY 850
      RETURN                                                            BODY 860
C                                                                       BODY 870
      END                                                               BODY 880
```

```
CBOND    0  10
      SUBROUTINE BOND (IGRP,IMR,RDIM1,M,N,DIAM,XDIM,L,RDIM2,IN)
C     SUBROUTINE TO CHECK BONC REQUIREMENTS USING ACI 318-71.
C
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),
     2    MTYPE(45),NGRF(45),NSPAC(6,45),NTIES(45)
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9),
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20),
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB
      INTEGER HEAD,DHEAD
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,
     4    NTIMES,NVEL,IINITO
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45),
     5   YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)
      IN=0
      IF ((L.EQ.20).OR.(M.EQ.1.AND.N.EQ.1)) GO TO 80
      MC=MCODE(IMR)
      MS=MBAR(IGRP,IMR)
      IF (MS.EQ.0.OR.MC.EQ.0) GO TO 80
      IF (L.LE.10) GO TO 20
      IF (L.EQ.11) GO TO 10
```

```
      IF (L.EQ.12) DL=.11E0*FCFY(MS)/SQRT(FCFY(MC))                     BOND 290
      GO TO 30                                                          BOND 300
10    DL=.085E0*FCFY(MS)/SQRT(FCFY(MC))                                 BOND 310
      GO TO 30                                                          BOND 320
20    DL=.01E0*(DIAM*DIAM)*PI*FCFY(MS)/SQRT(FCFY(MC))                   BOND 330
      TEMP=.0004E0*DIAM*FCFY(MS)                                        BOND 340
      IF (DL.LE.TEMP) DL=TEMP                                           BOND 350
30    IF (DL.LE.12.E0) DL=12.E0                                         BOND 360
      IF (RDIM1.GE.0.E0) DL=1.4E0*DL                                    BOND 370
      IF ((M.EQ.1).AND.(XDIM.LE.XBEGM(IMR)).AND.(DL.GE.RDIM2)) GO TO 40 BOND 380
      GO TO 60                                                          BOND 390
40    LINE=LINE+1                                                       BOND 400
      IF (LINE.GT.NL) CALL PAGE                                         BOND 410
50    FORMAT(27H **LONGITUDINAL REBAR GROUP,I3,12H OF ELEMENT ,I3,1H-,I3BOND 420
     1,47H HAS INSUFFICIENT DEVELOPMENT LENGTH FROM JOINT,I4,9H(BOND).**BOND 430
     2)                                                                 BOND 440
      PRINT 50, IGRP,IP(IMR),IQ(IMR),IP(IMR)                            BOND 450
      IREC=IREC+1                                                       BOND 460
      IN=1                                                              BOND 470
60    XEN=XDIM+RDIM2                                                    BOND 480
      XCOMP=XBEGM(IMR)+EFLM(IMR)                                        BOND 490
      IF ((N.EQ.1).AND.(XEN.GE.XCOMP).AND.(DL.GE.RDIM2)) GO TO 70       BOND 500
      GO TO 80                                                          BOND 510
70    LINE=LINE+1                                                       BOND 520
      IF (LINE.GT.NL) CALL PAGE                                         BOND 530
      WRITE(NPRT,50) IGRP,IP(IMR),IQ(IMR),IQ(IMR)                       BOND 540
      IN=1                                                              BOND 550
80    RETURN                                                            BOND 560
      END                                                               BOND 570
```

```
CCOEN    0  10
      SUBROUTINE COEN(M,UR,UD,IFLAG)                                    COEN   0
C                                                                       COEN  10
C THIS SUBROUTINE CALCULATES THE CONTRIBUTION OF CONFINED AND          COEN  20
C UNCONFINED CONCRETE TO THE TOTAL MEMBER ENERGY.                       COEN  30
C                                                                       COEN  40
      COMMON DATA(10000),KDATA(500)                                     COEN  50
      COMMON/CONBK/COAREA(4,45),SIGMA(5,45)                             COEN  60
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),       COEN  70
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),COEN  80
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)                      COEN  90
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9), COEN 100
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)COEN 110
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), COEN 120
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),      COEN 130
     2    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)     COEN 140
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), COEN 150
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      COEN 160
      INTEGER HEAD,DHEAD                                                COEN 170
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,COEN 180
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,COEN 190
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, COEN 200
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  COEN 210
     4    NTIMES,NVEL,IINITD                                            COEN 220
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),COEN 230
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),    COEN 240
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),COEN 250
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                COEN 260
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45), COEN 270
     5   YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)               COEN 280
```

```
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,
     1     LTABI,NMAX,NMAXI
      COMMON/STRNBK/SRP(4),SRQ(4),UX,UY,UZ,XLEN,AREA,ZZI,IMAT
C
      DIMENSION GPS(7,3),URMP(7,3),UDMP(7,3),GAUSS(3)
C INITIALIZE
      CON = COAREA(1,M)
      ISTAT=MSTAT(M)
      MTN = MATR(M)
      NG=NGRP(M)
      NT=NTIES(M)
      GAUSS(1)=5.E0/9.E0
      GAUSS(2)=8.E0/9.E0
      GAUSS(3)=GAUSS(1)
C RETRIEVE BEGINNING (ZERO) INDEXES FOR STRAIN AND STRESS HISTORIES.
      KRAIN=KDATA(LPI+M)+5*NG-1
      KRESS=KDATA(LPSI+M)+40*NG-1
C DETERMINE STRAINS AT GAUSS POINTS
      DO 20 I=1,7
      DO 20 J=1,3
      CALL STRN(M,XPI(J,M),YGP(I,M),GPS(I,J))
      IF (ABS(GPS(I,J)).LT.TINY) GPS(I,J) = SIGN(TINY,GPS(I,J))
20    CONTINUE
C IF NO STIRRUPS GO TO UNCONFINED SECTION.
      IF(NT.EQ.0) GO TO 190
C
C CALCULATE ENERGIES FOR CONFINED CONCRETE SECTION.
C
C     ESTABLISH POINTS 5 AND 6 OF STRESS-STRAIN CURVE.
      STS(5,MTN) = SIGMA(1,M)
```

```
      STN(5,MTN) = SIGMA(2,M)                                           COEN 590
      STN(6,MTN) = SIGMA(3,M)                                           COEN 600
      SLOPE(4,MTN) = SIGMA(4,M)                                         COEN 610
      SLOPE(5,MTN) = SIGMA(5,M)                                         COEN 620
      EPSU(MTN) = STN(4,MTN)                                            COEN 630
      IF(STS(5,MTN).LT.STS(4,MTN)) EPSU(MTN) = STN(5,MTN)               COEN 640
C                                                                       COEN 650
C EVALUATE STRESSES AND ENERGY DENSITIES AT GAUSS POINTS.               COEN 660
      KR=KRAIN+31                                                       COEN 670
      KS=KRESS+248                                                      COEN 680
       DO 150 I=1,3                                                     COEN 690
      K=I+2                                                             COEN 700
       DO 140 J=1,3                                                     COEN 710
       IF(ISTAT.EQ.3) GO TO 80                                          COEN 720
       DO 70 L=1,9                                                      COEN 730
70     S(L)=0.E0                                                        COEN 740
       GO TO 100                                                        COEN 750
80     S(9)=DATA(KR+J)                                                  COEN 760
       LS=KS+8*(J-1)                                                    COEN 770
       DO 90 L=1,8                                                      COEN 780
90    S(L)=DATA(LS+L)                                                   COEN 790
100    CALL CRET(MTN,GPS(K,J),URMP(K,J),UDMP(K,J))                      COEN 800
C UPDATE IF IFLAG=3                                                     COEN 810
       IF(IFLAG.NE.3.OR.ISTAT.NE.3) GO TO 120                           COEN 820
       DATA(KR+J)=S(9)                                                  COEN 830
       DO 110 L=1,8                                                     COEN 840
110    DATA(LS+L)=S(L)                                                  COEN 850
120    UR=UR+CON*GAUSS(I)*GAUSS(J)*URMP(K,J)                            COEN 860
       UD=UD+CON*GAUSS(I)*GAUSS(J)*UDMP(K,J)                            COEN 870
140    CONTINUE                                                         COEN 880
```

```
      KR=KR+3
150   KS=KS+24
C
C STORE MEMBER END STATE DATA IF IFLAG=3.
190  IF(IFLAG.NE.3.OR.ISTAT.NE.3) GO TO 200
     KR = KRAIN + 40
     KS = KRESS + 320
      XLOC=0.E0
      DO 180 I=1,2
      IF(I.EQ.2)XLOC=XLEN
      DO 170 KK=2,10
      K=KK-1
      CALL STRN(M,XLOC,YFIBR(KK,M),STRAIN)
     IF (ABS(STRAIN).LT.TINY) STRAIN = SIGN(TINY,STRAIN)
     S(9)=DATA(KR+K)
     LS=KS+8*(K-1)
     DO 155 L=1,8
155  S(L)=DATA(LS+L)
      CALL CRET(MTN,STRAIN,URE,UDE)
      DATA(KR+K)=S(9)
      DO 160 L=1,8
160   DATA(LS+L)=S(L)
170    CONTINUE
      KR=KR+9
180   KS=KS+72
C
C CALCULATE ENERGIES FOR UNCONFINED CONCRETE PORTION.
C
C EVALUATE STRESSES AND ENERGY DENSITIES AT GAUSS POINTS.
200   KR = KRAIN
```

```
      KS=KRESS                                                          COEN1190
      DO 270 I=1,7                                                      COEN1200
      DO 260 J=1,3                                                      COEN1210
      IF(ISTAT.EQ.3) GO TO 220                                          COEN1220
      DO 210 L=1,9                                                      COEN1230
210   S(L)=0.E0                                                         COEN1240
      GO TO 240                                                         COEN1250
220   S(9)=DATA(KR+J)                                                   COEN1260
      LS=KS+8*(J-1)                                                     COEN1270
      DO 230 L=1,8                                                      COEN1280
230   S(L)=DATA(LS+L)                                                   COEN1290
240   CALL CRET(MCODE(M),GPS(I,J),URMP(I,J),UDMP(I,J))                  COEN1300
C UPDATE IF IFLAG=3.                                                    COEN1310
      IF(IFLAG.NE.3.OR.ISTAT.NE.3) GO TO 260                            COEN1320
      DATA(KR+J)=S(9)                                                   COEN1330
     DO 250 L=1,8                                                       COEN1340
250   DATA(LS+L)=S(L)                                                   COEN1350
260   CONTINUE                                                          COEN1360
      KR=KR+3                                                           COEN1370
270    KS=KS+24                                                         COEN1380
C                                                                       COEN1390
C OBTAIN ENERGY CONTRIBUTIONS OF COVER AND SIDES.                       COEN1400
     DO 300 I=1,7                                                       COEN1410
     IF (I.LE.2) CONST = COAREA(3,M)                                    COEN1420
     IF (I.GE.6) CONST = COAPEA(4,M)                                    COEN1430
     IF (I.GE.3.AND.I.LE.5) CONST = COAREA(2,M)*GAUSS(I-2)              COEN1440
     DO 300 J=1,3                                                       COEN1450
     UR = UR + CONST*GAUSS(J)*URMP(I,J)                                 COEN1460
300  UD = UD + CONST*GAUSS(J)*UDMP(I,J)                                 COEN1470
C                                                                       COEN1480
```

```
C      UPDATE MEMBER END STATE DATA FOR UNCONFINED CONCRETE.                  COEN1490
        IF(IFLAG.NE.3.OR.ISTAT.NE.3) GO TO 360                                COEN1500
        KR=KRAIN+21                                                           COEN1510
       KS=KRESS+168                                                           COEN1520
        XLOC=0.E0                                                             COEN1530
       DO 350 I=1,2                                                           COEN1540
        IF(I.EQ.2)XLOC=XLEN                                                   COEN1550
        DO 340 KK=1,5                                                         COEN1560
       K=KK                                                                   COEN1570
       IF(KK.EQ.3) K=6                                                        COEN1580
       IF(KK.EQ.4) K=10                                                       COEN1590
       IF(KK.EQ.5) K=11                                                       COEN1600
       CALL STRN(M,XLOC,YFIBR(K,M),STRAIN)                                    COEN1610
       IF (ABS(STRAIN).LT.TINY) STRAIN = SIGN(TINY,STRAIN)                    COEN1620
       S(9)=DATA(KR+KK)                                                       COEN1630
       LS=KS+8*(KK-1)                                                         COEN1640
       DO 325 L=1,8                                                           COEN1650
325    S(L)=DATA(LS+L)                                                        COEN1660
        CALL CRET(MCODE(M),STRAIN,URE,UDE)                                    COEN1670
        DATA(KR+KK)=S(9)                                                      COEN1680
        DO 330 L=1,8                                                          COEN1690
330     DATA(LS+L)=S(L)                                                       COEN1700
340     CONTINUE                                                              COEN1710
       KR=KR+5                                                                COEN1720
350    KS = KS + 40                                                           COEN1730
360     RETURN                                                                COEN1740
C                                                                             COEN1750
C ***** GLOSSARY FOR COEN *****                                               COEN1760
C                                                                             COEN1770
C     KR = STRAIN INDEX FOR EACH INTEGRATION SECTION.                         COEN1780
```

```
C KRAIN = ZERO INDEX FOR STRAIN HISTORY STORAGE.                    COEN1790
C KRESS = ZERO INDEX FOR STRESS HISTORY STORAGE.                    COEN1800
C    KS = STRESS INDEX FOR EACH INTEGRATION SECTION.                COEN1810
C     M = MEMBER NUMBER.                                            COEN1820
C   MTN = MATERIAL NUMBER OF CONFINED CONCRETE.                     COEN1830
C    NG = NUMBER OF LONGITUDINAL STEEL GROUPS IN MEMBER.            COEN1840
C    NT = NUMBER OF STIRRUP GROUPS IN MEMBER.                       COEN1850
C    UD = DISSIPATIVE ENERGY DENSITY FOR MEMBER.                    COEN1860
C  UDMP = DISSIPATIVE ENERGY DENSITY AT MESH POINTS.                COEN1870
C    UR = RECOVERABLE ENERGY DENSITY FOR MEMBER.                    COEN1880
C  URMP = RECOVERABLE ENERGY DENSITY AT MESH POINTS.                COEN1890
C  XLEN = LENGTH OF MEMBER.                                         COEN1900
C                                                                   COEN1910
      END                                                           COEN1920
```

```
C
C     REMOVE CONCRETE AREA DISPLACED BY LONGITUDINAL STEEL.
      DO 20 N=1,NG
      NLOC = 1
      IF (YBAR(N,M).GT.YFIBR(2,M)) NLOC = 3
      IF (YBAR(N,M).LT.YFIBR(10,M)) NLOC = 4
20    COAREA(NLOC,M) = COAREA(NLOC,M) - AGRP(N,M)*XL(M)/4.E0
      IF (NTIES(M).NE.0) GO TO 30
      COAREA(2,M) = COAREA(2,M) + COAREA(1,M)
      COAREA(1,M) = 0.E0
      GO TO 70
C
30    MTN = MATR(M)
      MTS = MTIES(M)
      FCP = ABS(FCFY(MTN))
      SIGMA(1,M) = STS(5,MTN)
      SIGMA(2,M) = STN(5,MTN)
      SIGMA(3,M) = STN(6,MTN)
C
C     CHECK FOR INTERNALLY GENERATED CURVE.
      IF (STS(5,MTN).NE.0.E0) GO TO 40
C
C     ALTER POINTS 5 AND 6 ON THE STRESS-STRAIN CURVE.
      DSTS = ABS(0.75E0*PDF(7,MTS))
      IF (DSTS.GT.2.E3) DSTS = 2.E+03
      SIGMA(1,M) = STS(4,MTN) - DSTS
      DSTN = ABS(0.17E0*PDP(7,MTS))
      IF (DSTN.GT.8.E-3) DSTN = 8.E-03
      SIGMA(2,M) = STN(4,MTN) - DSTN
      S50 = (3.E0-2.E-3*FCP)/(FCP+1.E3) - ABS(0.75E0*PDP(7,MTS))
```

```
CCONC    0  10
      SUBROUTINE CONC (M)
C
C     THIS ROUTINE ESTABLISHES INTEGRATION CONSTANTS FOR CONCRETE, AND
C     ALTERS POINTS 5 AND 6 ON THE STRESS-STRAIN CURVE WHEN TIES ARE
C     WITHIN THE MEMBER.
C
      COMMON/CONBK/COAREA(4,45),SIGMA(5,45)
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),
     1     MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),
     2     MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9),
     1     SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20),
     1     PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BOM(10,45),
     1     BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),
     2     HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),
     3     TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),
     4     XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45),
     5     YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)
      INTEGER HEAD,DHEAD
C
C     COMPUTE AREAS OF CONCRETE SECTIONS (CONFINED,SIDES,TOP,BOTTOM).
      NG = NGRP(M)
      COAREA(1,M)  =  BPP(M)*DPP(M)*XL(M)/4.E0
      COAREA(2,M)  =  (BMEM(M)-BPP(M))*DPP(M)*XL(M)/4.E0
      COAREA(3,M)  =  BMEM(M)*ABS(YFIBR(1,M)-YFIBR(2,M))*XL(M)/4.E0
      COAREA(4,M)  =  BMEM(M)*ABS(YFIBR(10,M)-YFIBR(11,M))*XL(M)/4.E0
      IF (NG.EQ.0) GO TO 70
```

```
      SIGMA(3,M) = SIGMA(2,M) + (SIGMA(1,M)-STS(6,MTN))/
     1             (SIGMA(1,M)-0.5E0*FCP) * (S50-SIGMA(2,M))
C
C     CALCULATE SLOPES OF ALTERERED SEGMENTS OF STRESS-STRAIN CURVE.
40    CONTINUE
      CON4 = SIGMA(2,M) - STN(4,MTN)
      CON5 = SIGMA(3,M) - SIGMA(2,M)
      IF(ABS(CON4).LT.EPS) CON4 = SIGN(EPS,CON4)
      IF(ABS(CON5).LT.EPS) CON5 = SIGN(EPS,CON5)
      SIGMA(4,M) = (SIGMA(1,M)-STS(4,MTN))/CON4
      SIGMA(5,M) = (STS(6,MTN)-SIGMA(1,M))/CON5
70    CONTINUE
      RETURN
C
C ***** GLOSSARY FOR CONC *****
C
C COAREA = INTEGRATION CONSTANT OF CONCRETE CROSS-SECTION AREA.
C   CON4 = TEMPORARY CONSTANT OF STRESS DIFFERENCE.
C   CON5 = TEMPORARY CONSTANT OF STRESS DIFFERENCE.
C   NLOC = LOCATION OF COAREA (1=CONFINED CONCRETE,2=COVER ON THE
C          SIDES, 3=TOP COVER,4=BOTTOM COVER).
C  SIGMA = ALTERED STRESS, STRAINS, AND SLOPES DUE TO STIRRUPS IN EACH
C          MEMBER (1 = STRESS AT POINT 5 - 2,3 = STRAINS AT POINTS 5
C          AND 6 - 4,5 = SLOPES OF SEGMENTS  4 AND 5).
C
      END
```

```
CCRET    0  10
      SUBROUTINE CRET(MATL,RAINX,URC,UDC)
C
C DEFINITION OF STRESS HISTORY CURVE PARAMETERS
C TRANSMITTED VIA COMMON ARRAY S(9)
C
C S(1)= XFAIL = FIBER FAILURE CODE;0= NOT RUPTURED,1 = RUPTURED
C S(2) = SDIRCD= LOAD DIRECTICN CODE
C         0 = LOADING WITH STRAIN LT ULTIMATE
C         1 = UNLOADING WITH STRAIN LT ULTIMATE
C         2 = LOADING WITH STRAIN GT ULTIMATE
C         3 = UNLOADING WITH STRAIN GT ULTIMATE (DROP-ELASTIC)
C         4 = RELOADING WITH STRAIN GT ULTIMATE (DROP-ELASTIC)
C S(3) = RAINM = MAX STRAIN ENCOUNTERED
C S(4) = RESM = STRESS ASSOCIATED WITH RAINM
C S(5) = RAINL = STRAIN FROM LAST TIME STEP (LT PLASTIC OFFSET)
C S(6) = RESL = STRESS ASSOCIATED WITH RAINL
C S(7) = SSEGM = STRESS-STRAIN CURVE LINE SEGMENT CONTAINING (RAINM,RESM
C S(8) = PLAS = EFFECTIVE PLASTIC OFFSET STRAIN
C S(9) = RAIN = LAST STRAIN
C
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),
     $      XFAIL,SDIRCD,RAINM,RESM,RAINL,RESL,SSEGM,PLAS,RAIN,
     1     SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20),
     1     PI,PERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB
      INTEGER HEAD,DHEAD
C
C INITIALIZE.
C
```

```
      IF(XFAIL.GT.0.E0) GO TO 500
      J=MATL
      N=7
      YC = SLOPE(1,J)
      IF (SSEGM .LT. 1.E0) SSEGM= 1.E0
      SRAINM =RAINM
      SRESM = RESM
      DIRCD = SDIRCD
      SEGM = SSEGM
      RESX = RESL
      IF(SRAINM .EQ. 0.E0      )SRAINM = STN(1,J)
C
C DEFINE STRAIN ORIGIN
      ZERON = STN(1,J)+ PLAS
C
C TEST FOR TENSILE STRAIN OR STRAIN GT LAST PLASTIC OFFSET
      IF (RAINX .GT. ZERON) GO TO 450
C
C IF LAST STRAIN AND PRESENT STRAIN ARE EQUAL, COMPUTE ENERGIES.
      IF (RAINX .EQ. RAINL) GO TO 300
C
C TEST FOR UNLOADING OR RELOADING
      IF (RAINX .GE.RAINM) GO TO 200
C
C INCREASING STRAIN
C INCREMENT THRU STRESS-STRAIN CURVE TO LOCATE STRESS
C
      IPT= SSEGM +1.1E0
      DO 50  I=IPT,N
      IF(RAINX .LT. STN(I,J)) GO TO 50
```

```
      M= I-1                                                            CRET 590
      RESX = STS(M,J) + (RAINX - STN(M,J)) * SLOPE(M,J)                 CRET 600
      SEGM = M                                                          CRET 610
      GO TO 60                                                          CRET 620
   50 CONTINUE                                                          CRET 630
C                                                                       CRET 640
C STRAIN EXCEEDS RUPTURE                                                CRET 650
      XFAIL =1.E0                                                       CRET 660
      RESM=STS(N,J)                                                     CRET 670
      RAINM=STN(N,J)                                                    CRET 680
      SSEGM=N-1                                                         CRET 690
      RESL=0.E0                                                         CRET 700
      RAINL=RAINX                                                       CRET 710
      SDIRCD=DIRCD                                                      CRET 720
      GO TO 500                                                         CRET 730
   60 RAINM = RAINX                                                     CRET 740
      RESM  = RESX                                                      CRET 750
C                                                                       CRET 760
C SET LOAD DIRECTION CODE                                               CRET 770
      DIRCD = 0.E0                                                      CRET 780
      IF (RAINX .LT. EPSU(J)-EPS) DIRCD = 2.E0                          CRET 790
C                                                                       CRET 800
C CALCULATE PLASTIC OFFSET                                              CRET 810
      PLAS = RAINX - STN(1,J) - RESX/ YC                                CRET 820
      GO TO 300                                                         CRET 830
C                                                                       CRET 840
C UNLOADING OR RELOADING                                                CRET 850
C TEST FOR ASCENDING OR DESCENDING BRANCH                               CRET 860
C                                                                       CRET 870
  200 IF (SDIRCD .GT. 1.E0) GO TO 210                                   CRET 880
```

```
C                                                                       CRET 890
C ASCENDING BRANCH                                                      CRET 900
      DIRCD = 1.E0                                                      CRET 910
      RESX = RESM - (RAINM -RAINX)* YC                                  CRET 920
      GO TO 300                                                         CRET 930
C                                                                       CRET 940
C DESCENDING BRANCH TEST FOR UNLOADING OR RELOADING                     CRET 950
  210 DIRCD = 3.E0                                                      CRET 960
      IF ( RAINX .LT. RAINL) GO TO 220                                  CRET 970
C                                                                       CRET 980
C UNLOADING ON  DROP-ELASTIC LOOP                                       CRET 990
      RESX = UNLK(J)* YC * (RAINX - PLAS -STN(1,J))                     CRET1000
      GO TO 300                                                         CRET1010
C                                                                       CRET1020
C RELOADING ON DROP- ELASTIC LOOP                                       CRET1030
  220 DIRCD = 4.E0                                                      CRET1040
      RESX = RESM - UNLK(J)  * YC *(RAINM -RAINX)                       CRET1050
      GO TO 300                                                         CRET1060
C                                                                       CRET1070
C ENERGY CALCULATIONS                                                   CRET1080
C CHECK FOR UNLOADING ,RELOADING OR LOOP LOADING                        CRET1090
C                                                                       CRET1100
  300 IF ( SEGM .EQ. 1.E0 ) GO TO 350                                   CRET1110
      IF (DIRCD .EQ. 1.E0 ) GO TO 350                                   CRET1120
      IF( SDIRCD .EQ. 2.E0 .AND. DIRCD .EQ.3.E0) GO TO 360              CRET1130
      IF( SDIRCD .EQ. 3.E0.AND. DIRCD .EQ.4.E0) GO TO 370               CRET1140
      IF( SDIRCD .GE. 3.E0.AND. DIRCD .EQ.2.E0 ) GO TO 380              CRET1150
C                                                                       CRET1160
C DEFINE PREVIOUS UR FOR ASCENDING ^ DESCENDING BRANCHES                CRET1170
      UDC = 0.5E0 *SRESM*SRESM/ YC                                      CRET1180
```

```
      IF ( SDIRCD .GE. 2.E0 ) UDC = UDC* UNLK(J)                       CRET1190
C                                                                      CRET1200
C LOADING WITH INCREASING STRAIN                                       CRET1210
  305 URC = 0.5E0 * RESM * RESM/ YC                                    CRET1220
      IF (DIRCD .GT. 1.E0 ) URC= URC* UNLK(J)                          CRET1230
C                                                                      CRET1240
C ARE PRESENT ^ LAST STRAINS ON SAME LINE SEGMENT                      CRET1250
      IF (SSEGM .LT. SEGM) GO TO 310                                   CRET1260
C                                                                      CRET1270
C SAME SEGMENT                                                         CRET1280
      UDC = UDC + 0.5E0 *((SRESM + RESX) *( RAINX - SRAINM))           CRET1290
      UDC = UDC - URC                                                  CRET1300
      GO TO 400                                                        CRET1310
C                                                                      CRET1320
C ACCUMULATE AREA UNDER STRESS-STRAIN CURVE                            CRET1330
C LOCATE END OF LINE SEGMENT FOR PREVIOUS LOADING                      CRET1340
C                                                                      CRET1350
  310 IPT = SSEGM +1.1E0                                               CRET1360
C                                                                      CRET1370
C USE NO OF PRESENT SEGMENT                                            CRET1380
      N= SEGM                                                          CRET1390
      DO 330 I= IPT,N                                                  CRET1400
      UDC = UDC+ 0.5E0* ((SRESM+ STS(I,J)) *(STN(I,J)-SRAINM))         CRET1410
      SRESM = STS(I,J)                                                 CRET1420
  330 SRAINM = STN(I,J)                                                CRET1430
C                                                                      CRET1440
C ADD IN LAST SEGMENT                                                  CRET1450
      UDC= UDC + 0.5E0*(( STS(N,J)+ RESX) *( RAINX - STN(N,J)))        CRET1460
      UDC = UDC -URC                                                   CRET1470
      GO TO 400                                                        CRET1480
```

```
C
C UNLOADING   (ELASTIC RANGE)   0,1 ^ 1,1    LOADING ON FIRST SEGMENT
C
  350 UDC=0.E0
      URC= 0.5E0* RESX* RESX/ YC
      GO TO 400
C
C UNLOADING ON DROP-ELASTIC   2,3 ^ 3,3
  360 UDC = 0.E0
      URC = 0.5E0 * RESX * RESX / (UNLK(J)* YC)
      GO TO 400
C
C RELOADING WITHIN LOOP   3,4 ^ 4,4
  370 HT = (1.E0 - UNLK(J)) * RESM
      URC =  RESX - HT
      URC =   0.5E0 * URC * URC / (UNLK(J) * YC)
      UDC =   HT * (RAINX - RAINL)
      GO TO 400
C
C RELOADING FROM LOOP BACK ON TO VIRGIN CURVE   3,2 ^ 4,2
C OLD ELASTIC ENERGY
  380 UDC= 0.5E0 * UNLK(J)*SRESM*SRESM /YC
C
C ADDITIONAL LOOP  ENERGY
      HT= (1.E0 - UNLK(J))*SRESM
      UDC = UDC+ HT *(SRAINM - RAINL)
      GO TO 305
C
C SAVE CURVE PARAMETERS FOR NEXT CYCLE
C
```

```
  400 SSEGM = SEGM                                                      CRET1790
      RESL = RESX                                                       CRET1800
      RAINL= RAINX                                                      CRET1810
      SDIRCD= DIRCD                                                     CRET1820
      GO TO 600                                                         CRET1830
C                                                                       CRET1840
C SET DIRECTION CODE ^ STRESS FOR STRAIN GT PLASTIC OFFSET              CRET1850
  450 RESL = 0.E0                                                       CRET1860
      RAINL = ZERON                                                     CRET1870
      SDIRCD = 1.E0                                                     CRET1880
      IF  ( SRAINM .LT.EPSU(J)) SDIRCD = 3.E0                           CRET1890
C                                                                       CRET1900
C RETURN URC ^ UDC AS ZERO FOR TENSILE STRAIN OR CRUSHED CONCRETE.      CRET1910
  500 UDC= 0.E0                                                         CRET1920
      URC= 0.E0                                                         CRET1930
C                                                                       CRET1940
C REMEMBER STRAIN                                                       CRET1950
  600 RAIN = RAINX                                                      CRET1960
      RETURN                                                            CRET1970
C                                                                       CRET1980
C *****  GLOSSARY FOR CRET  *****                                       CRET1990
C                                                                       CRET2000
C  DIRCD = LOAD DIRECTICN CODE FOR THIS LOAD STEP (SEE SDIRCD ABOVE).   CRET2010
C     HT = LENGTH OF VERTICAL SEGMENT OF DROP-ELASTIC UNLOADING CURVE.  CRET2020
C    IPT = STRESS-STRAIN CURVE POINT NUMBER.                            CRET2030
C      J = MATERIAL NUMBER.                                             CRET2040
C      N = NUMBER OF POINTS USED TO DESCRIBE STRESS-STRAIN CURVE.       CRET2050
C   RESX = STRESS AT THIS LOAD STEP STRAIN (RAINX).                     CRET2060
C   SEGM = STRESS-STRAIN CURVE LINE SEGMENT NUMBER.                     CRET2070
C SRAINM = MAXIMUM STRAIN OF PREVIOUS LOAD STEP SAVED FOR               CRET2080
```

```
C           ENERGY CALCULATIONS.                                        CRET2090
C   SRESM = MAXIMUM STRESS OF PREVIOUS LOAD STEP SAVED FOR              CRET2100
C           ENERGY CALCULATIONS.                                        CRET2110
C      YC = ELASTIC MODULUS FOR CONCRETE.                               CRET2120
C   ZERON = EFFECTIVE STRAIN ORIGIN.                                    CRET2130
C                                                                       CRET2140
        END                                                             CRET2150
```

```
CCUTS    0  10
      SUBROUTINE CUTS
C
C     SUBROUTINE TO INCREASE THE NUMBER OF FINITE ELEMENTS DESCRIBED BY
C     INPUT DATA BY BISECTING ELEMENTS, DRAWN FROM COMMON BLOCKS.
C
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ (45),IQL(20),MATR(45),
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),
     2    MTYPE(45 ),NGRP(45),NSPAC(6,45),NTIES(45)
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50),
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),
     2    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)
      COMMON/LEADBK/AVOM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20),
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJO,NJER,NL,NLD,
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,
     4    NTIMES,NVEL,IINITD
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),
     1   BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),
     3    TFWF(45),TWWF(45),UOM(45),URM(45),XBEG(10,45),
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45),
     5   YFIBR(11,45),YLDS(45),XOM(45),PDF(7,45),DISM(45)
      INTEGER HEAD, DHEAD
      DIMENSION KIND(45)
      LDNM= NM
      DO 12 IT=1,NM
12    KIND(IT)=0
```

```
      EP=10.*SQRT(EPS)
C
C     JOINT SEARCH LOOP
C
      DO 40 I=1,NJ
      IF(NLS.EQ.0) GO TO 19
      DO 13 L=1,NLS
      IF(IPL(L).EQ.I.OR.IQL(L).EQ.I) GO TO 40
13    CONTINUE
C
C     ELEMENT SEARCH LOOP
C
19    NM=LDNM
      DO 20 J=1,NM
      IF(IP(J).EQ.I.OR.IQ(J).EQ.I) GO TO 40
20    CONTINUE
C
C     NEW JOINT, FIND BISECTED ELEMENT
C
      DO 30 J=1,NM
      I1=IP(J)
        I2=IQ(J)
      XLL = AMIN1(X(I1),X(I2))
      IF(X(I).LT.XLL) GO TO 30
      XR=AMAX1(X(I1),X(I2))
      IF(X(I).GT.XR) GO TO 30
      YB = AMIN1(Y(I1),Y(I2))
      IF(Y(I)+EP.LT.YB) GO TO 30
      YT=AMAX1(Y(I1),Y(I2))
      IF(Y(I)-EP.GT.YT) GO TO 30
```

```
      XDIF=X(I1)-X(I2)                                                  CUTS 590
      IF(ABS(XDIF).LE.EP ) GO TO 21                                     CUTS 600
      YDIF=Y(I1)-Y(I2)                                                  CUTS 610
      IF(ABS(YDIF).LE.EP) GO TO 21                                      CUTS 620
      SLOP =YDIF/XDIF                                                   CUTS 630
      XDIF=X(I1)-X(I)                                                   CUTS 640
       IF(ABS(XDIF).LE.EP) GO TO 30                                     CUTS 650
      YDIF=Y(I1)-Y(I)                                                   CUTS 660
      SNEW=YDIF/XDIF                                                    CUTS 670
      IF(ABS(SLOP-SNEW).GT.EP) GO TO 30                                 CUTS 680
C                                                                       CUTS 690
C     JOINT IS WITHIN ELEMENT J                                         CUTS 700
C                                                                       CUTS 710
21    LDNM=LDNM+1                                                       CUTS 720
C                                                                       CUTS 730
C     CHECK FOR NEED FOR EXCEEDING ARRAY SIZE                           CUTS 740
C                                                                       CUTS 750
      IF(LDNM.GT.NMD) GO TO 100                                         CUTS 760
        KIND(J)=1                                                       CUTS 770
      KIND(LDNM)=1                                                      CUTS 780
      IQ (LDNM)=IQ (J)                                                  CUTS 790
      IP(LDNM)=I                                                        CUTS 800
      IQ(J)=I                                                           CUTS 810
      XLEN=XL (J)                                                       CUTS 820
      XL(J)=SQRT((X(I1)-X(I))**2+(Y(I1)-Y(I))**2)                       CUTS 830
      XL(LDNM)=SQRT((X(I2)-X(I))**2+(Y(I2)-Y(I))**2)                    CUTS 840
      IF(MTYPE(J).EQ.4) GO TO 90                                        CUTS 850
C                                                                       CUTS 860
C     ELEMENTS PARAMETER CARD                                           CUTS 870
C                                                                       CUTS 880
```

```
      MSTAT(LDNM)=MSTAT(J )                                             CUTS 890
      MSHEAR(LDNM)=MSHEAR(J)                                            CUTS 900
      HTOP(LDNM)=HTOP(J)                                                CUTS 910
      D(LDNM)=D(J)                                                      CUTS 920
      DP(LDNM)=DP(J)                                                    CUTS 930
      R5=XLEN-EFLM(J)-XBEGM(J)                                          CUTS 940
      XBEGM(LDNM)=0.E0                                                  CUTS 950
      EFLM(LDNM)=XL(LDNM)-R5                                            CUTS 960
      EFLM(J)=XL(J)-XBEGM(J)                                            CUTS 970
C                                                                       CUTS 980
C     CONCRETE DATA CARD                                                CUTS 990
C                                                                       CUTS1000
      MCODE(LDNM)=MCODE(J)                                              CUTS1010
      MATR(LDNM)=MATR(J)                                                CUTS1020
      HMEM(LDNM)=HMEM(J)                                                CUTS1030
      BMEM(LDNM)=BMEM(J)                                                CUTS1040
      DPP(LDNM)=DPP(J)                                                  CUTS1050
      BPP(LDNM)=BPP(J)                                                  CUTS1060
C                                                                       CUTS1070
C     LONGTITUDIONAL REINFORCEMENT CARD                                 CUTS1080
C                                                                       CUTS1090
      ICNT=0                                                            CUTS1100
      KK=NGRP(J)                                                        CUTS1110
      DO 200 L=1,KK                                                     CUTS1120
      XEND=XLEN-EFFL(L,J)+X  (I1)                                       CUTS1130
      IF(X(I).LT.XEND) GO TO 80                                         CUTS1140
C                                                                       CUTS1150
C     REINFORCEMENT IS IN ONE SECTION                                   CUTS1160
C                                                                       CUTS1170
      EFFL(L,J)=X(I)-XEND                                               CUTS1180
```

```
      GO TO 200                                                         CUTS1190
C                                                                       CUTS1200
C     REINFORCEMENT IS BISECTED                                         CUTS1210
C                                                                       CUTS1220
80    ICNT=ICNT+1                                                       CUTS1230
      EFFL(ICNT,LDNM)=EFFL(L,J)                                         CUTS1240
      EFFL(L,J)=0 .E0                                                   CUTS1250
      XBEG(ICNT,LDNM)=0.E0                                              CUTS1260
      YBAR(ICNT,LDNM)=YBAR(L,J)                                         CUTS1270
      MBAR(ICNT,LDNM)=MBAR(L,J)                                         CUTS1280
      AGRP(ICNT,LDNM)=AGRP(L,J)                                         CUTS1290
200   CONTINUE                                                          CUTS1300
      NGRP(LDNM)=ICNT                                                   CUTS1310
C                                                                       CUTS1320
C     LATERAL REINFORCEMENT CARD                                        CUTS1330
C                                                                       CUTS1340
      KK=NTIES(J)                                                       CUTS1350
      ICNT=0                                                            CUTS1360
      MTIES(LDNM)=MTIES(J)                                              CUTS1370
      DO 300 L=1,KK                                                     CUTS1380
      XEND=NSPAC(L,J)*STIES(L,J)+XBEGS(L,J)+X(I1)                       CUTS1390
      IF(X(I).LT.XEND) GO TO 300                                        CUTS1400
C                                                                       CUTS1410
C     STIRRUP GROUP IS SUBDIVIDED                                       CUTS1420
C                                                                       CUTS1430
      KKK = NSPAC(L,J)                                                  CUTS1440
      ICNT=ICNT+1                                                       CUTS1450
      ATIES(ICNT,LDNM)=ATIES(L,J)                                       CUTS1460
      STIES(ICNT,LDNM)=STIES(L,J)                                       CUTS1470
      POP(ICNT,LDNM)=POP(L,J)                                           CUTS1480
```

```
      DO 322 II=1,KKK                                                   CUTS1490
      XB=X(I1)+XBEGS(L,J)+II*STIES(L,J)                                 CUTS1500
      IF(XB-EP.GT.X(I)) GO TO 323                                       CUTS1510
      IF(ABS(XB-X(I)).LT.EP) GO TO 324                                  CUTS1520
322    CONTINUE                                                         CUTS1530
      PRINT 325                                                         CUTS1540
325   FORMAT(1H ,11HCUTS BOMBED)                                        CUTS1550
324   II=II+1                                                           CUTS1560
323   XBEGS(ICNT,LDNM)=XB-X(I)                                          CUTS1570
      NSPAC(ICNT,LDNM)=KKK-II                                           CUTS1580
      NSPAC(L,J)=II-1                                                   CUTS1590
300   CONTINUE                                                          CUTS1600
      NTIES(LDNM)=ICNT                                                  CUTS1610
      GO TO 40                                                          CUTS1620
C                                                                       CUTS1630
C     WIDE FLANGE                                                       CUTS1640
C                                                                       CUTS1650
90    MATW(LDNM)=MATW(J)                                                CUTS1660
      MTYPE(LDNM)=MTYPE(J)                                              CUTS1670
      HTWF(LDNM)=HTWF(J)                                                CUTS1680
      DWF(LDNM)=DWF(J)                                                  CUTS1690
      TWWF(LDNM)=TWWF(J)                                                CUTS1700
      BWF(LDNM)=BWF(J)                                                  CUTS1710
      TFWF(LDNM)=TFWF(J)                                                CUTS1720
      GO TO 40                                                          CUTS1730
30    CONTINUE                                                          CUTS1740
      PRINT 35,I                                                        CUTS1750
35     FORMAT(1H ,9H***JOINT ,I3,45H IS NOT WITHIN ANY DEFINED ELEMENT  CUTS1760
     1(CUTS)***)                                                        CUTS1770
      IERR=IERR+1                                                       CUTS1780
```

```
40    CONTINUE                                             CUTS1790
      NM=LDNM                                              CUTS1800
      CALL ELIN(KIND)                                      CUTS1810
      RETURN                                               CUTS1820
100   PRINT 101                                            CUTS1830
101   FORMAT(1H ,32H***SPACE LIMIT EXCEEDED(CUTS)***)      CUTS1840
      IERR=IERR+1                                          CUTS1850
      RETURN                                               CUTS1860
      END                                                  CUTS1870
```

```
CDEFO    0  10
      SUBROUTINE DEFO (M)                                               DEFO   0
C                                                                       DEFO  10
C     THIS SUBROUTINE COMPUTES THE LOCAL DISTORTION VECTOR OF ELEMENT M DEFO  20
C                                                                       DEFO  30
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),       DEFO  40
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),DEFO  50
     2    MTYPE(45),NGRF(45),NSPAC(6,45),NTIES(45)                      DEFO  60
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), DEFO  70
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),      DEFO  80
     1    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)     DEFO  90
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), DEFO 100
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      DEFO 110
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT.IPRINT,DEFO 120
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,DEFO 130
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, DEFO 140
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  DEFO 150
     4    NTIMES,NVEL,IINITD                                            DEFO 160
      COMMON/STRNBK/SRP(4),SRQ(4),UX,UY,UZ,XLEN,AREA,ZZI,IMAT           DEFO 170
      INTEGER HEAD,DHEAD                                                DEFO 180
C                                                                       DEFO 190
C     INITIALIZE                                                        DEFO 200
C                                                                       DEFO 210
      IF (M.GT.0) GO TO 6                                               DEFO 220
      ILS=IABS(M)                                                       DEFO 230
      I=IPL(ILS)                                                        DEFO 240
      J=IQL(ILS)                                                        DEFO 250
      XLEN=1.E0                                                         DEFO 260
      GO TO 8                                                           DEFO 270
6     I=IP(M)                                                           DEFO 280
```

```
      J=IQ(M)                                                           DEFO 290
8     DX1=(X(J)-X(I))/XLEN                                              DEFO 300
      DX2=(Y(J)-Y(I))/XLEN                                              DEFO 310
      DU1=(XDJ(1,J)-XDJ(1,I))*AVGL/XLEN                                 DEFO 320
      DU2=(XDJ(2,J)-XDJ(2,I))*AVGL/XLEN                                 DEFO 330
      UI3=XDJ(3,I)                                                      DEFO 340
      DU3=XDJ(3,J)-UI3                                                  DEFO 350
C                                                                       DEFO 360
C     TEST FOR GEOMETRIC NONLINEARITY                                   DEFO 370
C                                                                       DEFO 380
      IF (ILIN.GT.0) GO TO 10                                           DEFO 390
C                                                                       DEFO 400
C     GEOMETRICALLY LINEAR (ILIN=0)                                     DEFO 410
C     DISTORTIONS OF MEMBER M IN GLOBAL COORDINATES (TRANSLATIONAL)     DEFO 420
C                                                                       DEFO 430
      D1=UI3*DX2+DU1                                                    DEFO 440
      D2=-UI3*DX1+DU2                                                   DEFO 450
      GO TO 20                                                          DEFO 460
C                                                                       DEFO 470
C     GEOMETRICALLY NONLINEAR: UNLIMITED RIGID-BODY MOTIONS,BEAM-COLUMN DEFO 480
C     DISTORTIONS (ILIN=1)                                              DEFO 490
C     ROTATIONAL TRANSFORMATICN PARAMETERS: GLOBAL TO DEFORMED JOINT I  DEFO 500
C                                                                       DEFO 510
10    SI3=SIN(UI3)                                                      DEFO 520
      CI3=COS(UI3)                                                      DEFO 530
      SI32=SIN(UI3/2.E0)**2                                             DEFO 540
C                                                                       DEFO 550
C     DISTORTIONS OF MEMBER M IN JOINT-I COORDINATES (TRANSLATIONAL)    DEFO 560
C                                                                       DEFO 570
      D1=-2.E0*SI32*DX1+SI3*(DX2+DU2)+CI3*DU1                           DEFO 580
```

```
      D2=-SI3*(DX1+DU1)-2.E0*SI32*DX2+CI3*DU2                           DEFO 590
C                                                                       DEFO 600
C     ROTATIONAL TRANSFORMATION PARAMETERS: GLOBAL TO LOCAL (UNDEFORMED DEFO 610
C     MEMBER) CONVENTION  FOR MEMBER AXES: 1-AXIS FROM JOINT I TO J,3-AXDEFO 620
C     WITH SAME SENSE AS 3-GLOBAL, 2-AXIS TO FORM A RIGHT-HANDED TRIAD. DEFO 630
C                                                                       DEFO 640
20    C3=DX1                                                            DEFO 650
      S3=DX2                                                            DEFO 660
C                                                                       DEFO 670
C     ROTATIONAL TRANSFORMATION : GLOBAL TO INITIAL LOCAL.              DEFO 680
C     DISTORTIONS ARE PER UNIT LENGTH OF MEMBER M IN LOCAL COORDINATES. DEFO 690
C                                                                       DEFO 700
      UX=C3*D1+S3*D2                                                    DEFO 710
      UY=-S3*D1+C3*D2                                                   DEFO 720
      UZ=DU3                                                            DEFO 730
      RETURN                                                            DEFO 740
      END                                                               DEFO 750
```

```
CDELT    0  10
      SUBROUTINE DELT (SOLN,DELTA,I,DVALUE,ISEQ)                        DELT   0
C                                                                       DELT  10
C     THIS SUBROUTINE CALCULATES THE DERIVATIVE OF THE POTENTIAL FUNCTIODELT  20
C     (DVALUE) FOR A SMALL CHANGE (DELTA) IN THE I-TH VARIABLE OF THE   DELT  30
C     SOLUTION ARRAY.  FOR MOST STRUCTURAL MODELS A CHANGE IN A SINGLE  DELT  40
C     DEGREE-OF-FREEDOM CAUSES ONLY A FEW OF THE MANY TERMS CONTRIBUTINGDELT  50
C     TO THE POTENTIAL FUNCTION TO CHANGE IN VALUE.  THE CODING TAKES   DELT  60
C     ADVANTAGE OF THIS BY IDENTIFYING AND THEN EVALUATING ONLY THOSE   DELT  70
C     TERMS AFFECTED BY THE I-TH VARIABLE.                              DELT  80
C                                                                       DELT  90
C     THE PARAMETER (ISEQ) CONTROLS INITIALIZATION OF THE CALCULATIONS  DELT 100
C     AS FOLLOWS                                                        DELT 110
C     ISEQ=0, IT IS ASSUMED THAT THE SUBROUTINE IS BEING CALLED FOR     DELT 120
C           SEQUENTIAL VALUES OF I FROM 1 TO NDFJ.  THUS THE 2-D        DELT 130
C           DISPLACEMENT ARRAYS ARE FORMED FOR I=1 ONLY.                DELT 140
C     ISEQ=1, INDICATES THAT THE SUBROUTINE IS BEING CALLED WITH THE    DELT 150
C           PREVIOUS VALUE FOR I BUT A NEW STEP SIZE (DELTA).  IN       DELT 160
C           THIS CASE CERTAIN CALCULATIONS ARE NOT REPEATED.            DELT 170
C     ISEQ=2, INDICATES THAT THE 2-D DISPLACEMENT ARRAYS HAVE BEEN      DELT 180
C           FORMED.  THUS THIS CALCULATION MAY BE SKIPPED.              DELT 190
C                                                                       DELT 200
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),       DELT 210
     1    MATH(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),DELT 220
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)                      DELT 230
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), DELT 240
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),      DELT 250
     1    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)     DELT 260
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), DELT 270
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      DELT 280
```

```
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,DELT 290
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,DELT 300
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, DELT 310
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  DELT 320
     4    NTIMES,NVEL,IINITD                                            DELT 330
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),DELT 340
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),    DELT 350
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),DELT 360
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                DELT 370
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45), DELT 380
     5   YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)               DELT 390
C                                                                       DELT 400
      INTEGER HEAD,DHEAD                                                DELT 410
      DIMENSION SOLN(90)                                                DELT 420
C                                                                       DELT 430
      IF(ISEQ.EQ.1) GO TO 50                                            DELT 440
      IF (ISEQ.EQ.2) GO TO 30                                           DELT 450
C                                                                       DELT 460
C     INITIALIZE.                                                       DELT 470
C                                                                       DELT 480
      IF (I.NE.1) GO TO 30                                              DELT 490
      DO 20 N=1,NDFJ                                                    DELT 500
      J=IDFI(N)                                                         DELT 510
      K=IDFII(N)                                                        DELT 520
20    XDJ(K,J)=SOLN(N)                                                  DELT 530
      L=NDFJ+1                                                          DELT 540
      DO 16 N=L,NDF                                                     DELT 550
      M=IDFI(N)                                                         DELT 560
16    XDM(M)=SOLN(N)                                                    DELT 570
30    J=IDFI(I)                                                         DELT 580
```

```
      K=IOFII(I)                                                        DELT 590
C                                                                       DELT 600
C     CALCULATE CONTRIBUTIONS TO THE DERIVATIVE FROM THE MASSES AND     DELT 610
C     FORCING FUNCTIONS.                                                DELT 620
C                                                                       DELT 630
      VALUE1=0.E0                                                       DELT 640
      VALUE2=0.E0                                                       DELT 650
      IF(I.GT.NDFJ)GO TO 71                                             DELT 660
      IF (DT.EQ.0.E0.OR.DAS(K,J).EQ.0.E0) GO TO 40                      DELT 670
      VALUE1=DAS(K,J)*(6.E0*(SOLN(I)-DIS(K,J))/(DT*DT)-6.E0*VEL(K,J)/DT-DELT 680
     12.E0*ACC(K,J))                                                    DELT 690
40    IF (IFOR.EQ.0.AND.NFF.EQ.0) GO TO 50                              DELT 700
      VALUE2=-F(K,J)                                                    DELT 710
C                                                                       DELT 720
C     CALCULATE MEMBER ENERGY FOR THE I-TH VARIABLE WITH NEGATIVE INCREMDELT 730
C                                                                       DELT 740
50    IF(I.GT.NDFJ) GO TO 71                                            DELT 750
      VALUE3=0.E0                                                       DELT 760
      VALUE4=0.E0                                                       DELT 770
      IF (XDJ(K,J).EQ.0.E0) GO TO 60                                    DELT 780
      DELXDJ=DELTA/XDJ(K,J)                                             DELT 790
      IF (ABS(DELXDJ).GT.EPS) GO TO 60                                  DELT 800
      DVALUE=0.E0                                                       DELT 810
      GO TO 90                                                          DELT 820
60    XDJ(K,J)=XDJ(K,J)-DELTA                                           DELT 830
      DO 70 L=1,NM                                                      DELT 840
      IF (IP(L).NE.J.AND.IQ(L).NE.J) GO TO 70                           DELT 850
      CALL ENGY (L,UR,UD)                                               DELT 860
      VALUE3=VALUE3+UR                                                  DELT 870
      VALUE4=VALUE4+UD                                                  DELT 880
```

```
70    CONTINUE                                                          DELT 890
      GO TO 73                                                          DELT 900
71    IF(XDM(J).EQ.0.E0)GO TO 72                                        DELT 910
      IF(ABS(DELT I/XDM(J)).GT.EPS)GO TO 72                             DELT 920
      DVALUE=0.E0                                                       DELT 930
      GO TO 90                                                          DELT 940
72    XDM(J)=XDM(J)-DELTA                                               DELT 950
      CALL ENGY(J,VALUE3,VALUE4)                                        DELT 960
      GO TO 76                                                          DELT 970
73    IF (NLS.LE.0) GO TO 76                                            DELT 980
      DO 74 M=1,NLS                                                     DELT 990
      IF(IPL(M).NE.J.AND.IQL(M).NE.J) GO TO 74                          DELT1000
      CALL LEAF(-M,UR,1)                                                DELT1010
      VALUE3=VALUE3+UR                                                  DELT1020
74    CONTINUE                                                          DELT1030
C                                                                       DELT1040
C     CALCULATE MEMBER ENERGY FOR THE I-TH VARIABLE WITH POSITIVE INCREMDELT1050
C                                                                       DELT1060
76    DELTA2=DELTA+DELTA                                                DELT1070
      IF(I.GT.NDFJ) GO TO 87                                            DELT1080
      VALUE5=0.E0                                                       DELT1090
      VALUE6=0.E0                                                       DELT1100
      XDJ(K,J)=XDJ(K,J)+DELTA2                                          DELT1110
      DO 80 L=1,NM                                                      DELT1120
      IF (IP(L).NE.J.AND.IQ(L).NE.J) GO TO 80                           DELT1130
      CALL ENGY (L,UR,UD)                                               DELT1140
      VALUE5=VALUE5+UR                                                  DELT1150
      VALUE6=VALUE6+UD                                                  DELT1160
80    CONTINUE                                                          DELT1170
      IF (NLS.EQ.0) GO TO 86                                            DELT1180
```

```
      DO 84 M=1,NLS                                                     DELT1190
      IF (IPL(M).NE.J.AND.IQL(M).NE.J) GO TO 84                         DELT1200
      CALL LEAF(-M,UR,1)                                                DELT1210
      VALUE5=VALUE5+UR                                                  DELT1220
84    CONTINUE                                                          DELT1230
86    XDJ(K,J)=SOLN(I)                                                  DELT1240
      GO TO 89                                                          DELT1250
87    XDM(J)=XDM(J)+DELTA2                                              DELT1260
      CALL ENGY(J,VALUE5,VALUE6)                                        DELT1270
      XDM(J)=SOLN(I)                                                    DELT1280
C                                                                       DELT1290
C     CALCULATE DERIVATIVE.                                             DELT1300
C                                                                       DELT1310
89    DVALUE=(VALUE1+VALUE2+(VALUE5+VALUE6-VALUE3-VALUE4)/DELTA2)/AVDM  DELT1320
90    RETURN                                                            DELT1330
      END                                                               DELT1340
```

```
CELIN   0  10
      SUBROUTINE ELIN(KIND)                                             ELIN   0
C                                                                       ELIN  10
C     SUBROUTINE TO PRINT A TABLE OF ELEMENT CHARACTERISTICS TAKEN      ELIN  20
C     FROM COMMON BLOCKS.                                               ELIN  30
C                                                                       ELIN  40
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ (45),IQL(20),MATR(45),      ELIN  50
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),ELIN  60
     2    MTYPE(45 ),NGRP(45),NSPAC(6,45),NTIES(45)                     ELIN  70
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9), ELIN  80
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)ELIN  90
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,ELIN 100
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,ELIN 110
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, ELIN 120
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  ELIN 130
     4    NTIMES,NVEL,IINITD                                            ELIN 140
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BOM(10,45),ELIN 150
     1   BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),     ELIN 160
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),ELIN 170
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                ELIN 180
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45), ELIN 190
     5   YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)               ELIN 200
      DIMENSION KIND(45)                                                ELIN 210
      DATA RC/3HR/C/,WF/3HW/F/,DET/6HDETAIL/,BIS/6HBISECT/,ED/2HED/     ELIN 220
      CALL PAGE                                                         ELIN 230
      WRITE(NPRT,30)                                                    ELIN 240
30    FORMAT(1H0,18X,51HSUMMARY DATA FOR FINITE ELEMENTS USED IN SIMULATELIN 250
     1ION)                                                              ELIN 260
      LINE=LINE+2                                                       ELIN 270
      WRITE(NPRT,40)                                                    ELIN 280
```

```
40     FORMAT(1H0,6X,7HELEMENT,6X,6HJOINTS,9X,4HKIND,8X,4HTYPE,12X,6HLENGELIN 290
      1TH,12X,8HMATERIAL/)                                             ELIN 300
       LINE=LINE+3                                                     ELIN 310
        DO 100 I=1,NM                                                  ELIN 320
       LINE=LINE+1                                                     ELIN 330
        IF(LINE.LE.NL) GO TO 50                                        ELIN 340
        CALL PAGE                                                      ELIN 350
        WRITE(NPRT,40)                                                 ELIN 360
       LINE=LINE+3                                                     ELIN 370
50     TTYPE=RC                                                        ELIN 380
        AKIND=DET                                                      ELIN 390
       IF(KIND(I).EQ.1) AKIND=BIS                                      ELIN 400
       IF(MTYPE(I).EQ.4) GO TO 70                                      ELIN 410
       MATL=MCODE(I)                                                   ELIN 420
       GO TO 80                                                        ELIN 430
70      TTYPE=WF                                                       ELIN 440
       MATL=MATW(I)                                                    ELIN 450
80     WRITE(NPRT,90)I,IP(I),IG(I),AKIND,ED,TTYPE,XL(I),NAME(MATL)     ELIN 460
90      FORMAT(1H ,8X,I3,6X,I3,1H-,I3,8X,A6,A2,7X,A3,8X,E14.7,10X,A4)  ELIN 470
100    CONTINUE                                                        ELIN 480
       RETURN                                                          ELIN 490
       END                                                             ELIN 500
```

```
CENDS    0  10
      SUBROUTINE ENDS (IMR,M,N,XDIM,RDIM1,RDIM2,DIAM)
C
C     SUBROUTINE TO CALCULATE EFFECTIVE LENGTH OF REBAR.
C
      COMMON /MEMBER/ DM1(1305),D(45),DP(45),DM2(585),HMEM(45),DM3(3745)
C
      DEP=D(IMR)
      IF(RDIM1.LT.0.E0) DEP = HMEM(IMR) -DP(IMR)
      TEMP=12.E0*DIAM
      IF (DEP.GT.TEMP) TEMP=DEP
      IF (M.EQ.1) GO TO 10
      XDIM=XDIM+TEMP
      RDIM2=RDIM2-TEMP
10    IF (N.EQ.1) GO TO 20
      RDIM2=RDIM2-TEMP
20    RETURN
      END
```

```
CENGY    0  10
      SUBROUTINE ENGY (M,UR,UD)                                         ENGY   0
C                                                                       ENGY  10
C     THIS SUBROUTINE MANAGES THE RECOVERABLE STRAIN ENERGY (UR) AND    ENGY  20
C     THE DISSIPATIVE STRAIN ENERGY (UD) CALCULATION FOR MEMBER (M).    ENGY  30
C     THE TYPE OF MEMBER IS INDICATED BY MTYPE(M), WHERE                ENGY  40
C     MTYPE=1, INDICATES A REINFORCED CONCRETE MEMBER WITH STIRRUPS.    ENGY  50
C     MTYPE=2, INDICATES A REINFORCED CONCRETE MEMBER WITH TIES.        ENGY  60
C     MTYPE=3, INDICATES A REINFORCED CONCRETE MEMBER WITH AN           ENGY  70
C              EMBEDDED WIDE FLANGE.                                    ENGY  80
C     MTYPE=4, INDICATES A NONCOMPOSITE WIDE FLANGE MEMBER.             ENGY  90
C     MTYPE=5, INDICATES A LINEAR LEAF SPRING MEMBER.                   ENGY 100
C                                                                       ENGY 110
      COMMON/ELEMET/IDUM1(851),MTYPE(45),IDUM2(360)                     ENGY 120
C                                                                       ENGY 130
C     GENERAL REINFORCED CONCRETE MEMBER.                               ENGY 140
      IF (MTYPE(M).LT.4) CALL MEMB (M,UR,UD,1)                          ENGY 150
C                                                                       ENGY 160
C     STEEL WIDE FLANGE MEMBER.                                         ENGY 170
      IF (MTYPE(M).EQ.4) CALL WIDE (M,UR,UD,1)                          ENGY 180
      RETURN                                                            ENGY 190
      END                                                               ENGY 200
```

```
CERRS    0  10
      SUBROUTINE ERRS (SOLN,VALUEM)
C
C     THIS SUBROUTINE CALCULATES THE ERROR MEASURES AT THE MIDDLE AND EN
C     OF THE TIME INTERVAL.
C
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50),
     1     ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),
     1     XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20),
     1     PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,
     1     IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,
     2     NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,
     3     NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,
     4     NTIMES,NVEL,IINITD
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),
     1     BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),
     2     HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),
     3     TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),
     4     XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45),
     5    YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)
      COMMON/SEEKBK/DEFOR(90),STPSIZ(90),GRAD(90),GRADI(90),DELTAG(90),
     1     DIRECT(90),DIAG(90),STEP(4),DSTEP(4),FVAL(4),VALUES(7),
     2     DISACC,SSIZE,FUNACC,FUNMIN,CRITL,CRITU,NLIN
C
      DIMENSION SOLN(90)
      INTEGER HEAD,DHEAD
C
C
```

```
C     CALCULATE ERROR AT END OF TIME INTERVAL.                                  ERRS 290
      RERF=0.E0                                                                 ERRS 300
      DO 10 I=1,NDFJ                                                            ERRS 310
      J=IDFI(I)                                                                 ERRS 320
      K=IDFII(I)                                                                ERRS 330
      DELTA=STPSIZ(I)                                                           ERRS 340
      CALL DELT (SOLN,DELTA,I,DER(K,J),2)                                       ERRS 350
      IF (VALUEM.EQ.0.E0) VALUEM=TINY                                           ERRS 360
      RESENG(K,J)=DER(K,J)*SOLN(I)                                              ERRS 370
      ERJF(K,J)=RESENG(K,J)/VALUEM                                              ERRS 380
      RERF=AMAX1(ABS(ERJF(K,J)),RERF)                                           ERRS 390
10    CONTINUE                                                                  ERRS 400
C                                                                               ERRS 410
C     IF STATIC ANALYSIS, RETURN                                                ERRS 420
C                                                                               ERRS 430
      IF (DT.EQ.0.E0.OR.NMAS.EQ.0) GO TO 70                                     ERRS 440
C                                                                               ERRS 450
C     CALCULATE DISPLACEMENTS AT MIDDLE OF TIME INTERVAL.                       ERRS 460
C                                                                               ERRS 470
      DO 30 I=1,NDFJ                                                            ERRS 480
      J=IDFI(I)                                                                 ERRS 490
      K=IDFII(I)                                                                ERRS 500
      IF (DAS(K,J).EQ.0.E0) GO TO 20                                            ERRS 510
      BETX=6.E0*((SOLN(I)-DIS(K,J))/(DT*DT)-VEL(K,J)/DT-0.5E0*ACC(K,J))/ERRS 520
     1DT                                                                        ERRS 530
      DEFOR(I)=BETX*DT*DT*DT/48.E0+ACC(K,J)*DT*DT/8.E0+VEL(K,J)*DT/2.E0+ERRS 540
     1DIS(K,J)                                                                  ERRS 550
      GO TO 30                                                                  ERRS 560
20    DEFOR(I)=(SOLN(I)+DIS(K,J))/2.E0                                          ERRS 570
30    CONTINUE                                                                  ERRS 580
```

```
      L=NDFJ+1
      DO 35 I=L,NDF
      M=IDFI(I)
35    DEFOR(I)=(SOLN(I)+DISM(M))/2.E0
C
C     CALCULATE FORCING FUNCTIONS AT MIDDLE OF TIME INTERVAL.
C
      SAVE1=DT
      SAVE2=TIME
      DT=DT/2.E0
      TIME=TIME-DT
      IF (NFF.EQ.0) GO TO 40
      CALL FORS (TIME)
C
C     CALCULATE ERROR AT MIDDLE OF TIME INTERVAL.
C
40    RERH=0.E0
      DO 50 I=1,NDFJ
      J=IDFI(I)
      K=IDFII(I)
      IF (DAS(K,J).EQ.0.E0) GO TO 50
      DELTA=STPSIZ(I)
      CALL DELT (DEFOR,DELTA,I,DERIV,0)
      ERJH(K,J)=DERIV*DEFOR(I)/VALUEM
      RERH=AMAX1(ABS(ERJH(K,J)),RERH)
50    CONTINUE
C
C     RESET DATA
C
      DT=SAVE1
```

```
      TIME=SAVE2                                                        ERRS 890
      DO 60 I=1,NDFJ                                                    ERRS 900
      J=IDFI(I)                                                         ERRS 910
      K=IDFII(I)                                                        ERRS 920
   60 XDJ(K,J)=SOLN(I)                                                  ERRS 930
      L=NDFJ+1                                                          ERRS 940
      DO 65 I=L,NDF                                                     ERRS 950
      M=IDFI(I)                                                         ERRS 960
   65 XDM(M)=SOLN(I)                                                    ERRS 970
   70 RETURN                                                            ERRS 980
      END                                                               ERRS 990
```

```
CFAIL    0  10
      SUBROUTINE FAIL                                                   FAIL   0
C                                                                       FAIL  10
C     THIS SUBROUTINE CHECKS THE MACROSCOPIC FAILURE CRITERIA FOR       FAIL  20
C     A REINFORCED CONCRETE MEMBER(M).                                  FAIL  30
C                                                                       FAIL  40
C                                                                       FAIL  50
      COMMON DATA(10000),KDATA(500)                                     FAIL  60
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),       FAIL  70
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),FAIL  80
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)                      FAIL  90
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9), FAIL 100
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)FAIL 110
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), FAIL 120
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      FAIL 130
      INTEGER HEAD,DHEAD                                                FAIL 140
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,FAIL 150
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,FAIL 160
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, FAIL 170
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  FAIL 180
     4    NTIMES,NVEL,IINITD                                            FAIL 190
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),FAIL 200
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),    FAIL 210
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),FAIL 220
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                FAIL 230
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45), FAIL 240
     5   YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)               FAIL 250
      COMMON/SAVEBK/SAVACC(3,50),SAVAXL(2,45),SAVCRV(2,45),SAVMOM(2,45) FAIL 260
     1    ,SAVSHR(2,45),SAVSRP(3,20),SAVSRQ(3,20),SAVXDJ(3,50),         FAIL 270
     2    SAVVEL(3,50),SVSTRN(12,45),SVSTRS(12,45)                      FAIL 280
```

```
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,          FAIL 290
     1    LTABI,NMAX,NMAXI                                              FAIL 300
      COMMON/STRNBK/SRP(4),SRQ(4),UX,UY,UZ,XLEN,AREA,ZZI,IMAT           FAIL 310
      LOGICAL PASS(2)                                                   FAIL 320
      DIMENSION ASUBV(2),F(2),FFOUR(2),FTWO(2)                          FAIL 330
     1 ,TEMP(24),STRESS(24) ,STRAIN(24),SFIVE(24),SEIGHT(24)            FAIL 340
C                                                                       FAIL 350
      LINE = NL                                                         FAIL 360
      EGSIS = 1.E0                                                      FAIL 370
      IF (IUNITS.EQ.1.OR.IUNITS.EQ.2) EGSIS = 6894.75789                FAIL 380
      DO 7000 M=1,NM                                                    FAIL 390
      DO 3000 L=1,2                                                     FAIL 400
      SAVAXL(L,M) = 0.E0                                                FAIL 410
      SAVCRV(L,M) = 0.E0                                                FAIL 420
      SAVMOM(L,M) = 0.E0                                                FAIL 430
      SAVSHR(L,M) = 0.E0                                                FAIL 440
3000  CONTINUE                                                          FAIL 450
      XLEN = XL(M)                                                      FAIL 460
      CALL DEFO(M)                                                      FAIL 470
C                                                                       FAIL 480
C     VERIFY ELEMENT TYPE                                               FAIL 490
      IF(MTYPE(M).EQ.5) GO TO 7000                                      FAIL 500
      IF(MTYPE(M).EQ.4) GO TO 3170                                      FAIL 510
      CALL MEMB(M,UR,UD,3)                                              FAIL 520
      NGRPM = NGRP(M)                                                   FAIL 530
      NNG = NGRPM + 14                                                  FAIL 540
      NTIESM = NTIES(M)                                                 FAIL 550
C                                                                       FAIL 560
C     COLLECT ELEMENT DIMENSIONS                                        FAIL 570
      TEMP(1) = YFIBR( 1,M)                                             FAIL 580
```

```
      TEMP(2) = YFIBR( 2,M)                                              FAIL 590
      TEMP(3) = YFIBR( 6,M)                                              FAIL 600
      TEMP(4) = YFIBR(10,M)                                              FAIL 610
      TEMP(5) = YFIBR(11,M)                                              FAIL 620
      DO 3010 J=6,14                                                     FAIL 630
      TEMP(J) = YFIBR(J-4,M)                                             FAIL 640
 3010 CONTINUE                                                           FAIL 650
      DO 3020 J=15,24                                                    FAIL 660
      TEMP(J) = YBAR(J-14,M)                                             FAIL 670
 3020 CONTINUE                                                           FAIL 680
      DO 3150 L=1,2                                                      FAIL 690
      ESMAX = -TINY                                                      FAIL 700
      ESMIN = TINY                                                       FAIL 710
      NESMAX = 1                                                         FAIL 720
      NESMIN = 1                                                         FAIL 730
      IF(MSTAT(M).EQ.3) GO TO 3040                                       FAIL 740
C                                                                        FAIL 750
C     STRESS AND STRAIN COMPUTATIONS FOR AN ELASTIC ELEMENT              FAIL 760
      XLOC = 0.E0                                                        FAIL 770
      IF(L.EQ.2) XLOC = XLEN                                             FAIL 780
      DO 3030 K=1,NNG                                                    FAIL 790
      IF(K.GE.1.AND.K.LE.5) J = MCODE(M)                                 FAIL 800
      IF(K.GE.6.AND.K.LE.14) J = MATR(M)                                 FAIL 810
      IF(K.GE.15.AND.K.LE.24) J = MBAR(K-14,M)                           FAIL 820
      CALL STRN(M,XLOC,TEMP(K),STRAIN(K))                                FAIL 830
      ELASMD = ET(J)                                                     FAIL 840
      IF(STRAIN(K).LE.0.E0) ELASMD = EC(J)                               FAIL 850
      STRESS(K) = ELASMD*STRAIN(K)                                       FAIL 860
      IF(K.LE.14.AND.STRAIN(K).GT.0.E0) STRESS(K) = 0.E0                 FAIL 870
      IF(K.LT.15) GO TO 3030                                             FAIL 880
```

```
      IF(STRAIN(K).LT.ESMAX) GO TO 3025                                 FAIL 890
      ESMAX=STRAIN(K)                                                   FAIL 900
      NESMAX=K-14                                                       FAIL 910
      GO TO 3030                                                        FAIL 920
 3025 IF(STRAIN(K).GT.ESMIN) GO TO 3030                                 FAIL 930
      ESMIN=STRAIN(K)                                                   FAIL 940
      NESMIN=K-14                                                       FAIL 950
 3030 CONTINUE                                                          FAIL 960
      GO TO 3100                                                        FAIL 970
C                                                                       FAIL 980
C     STRESSES AND STRAINS FOR YIELDED R/C ELEMENTS                     FAIL 990
C                                                                       FAIL1000
 3040 INDEX = KDATA(LPSI+M) -1+NGRPM*40+168                             FAIL1010
C                                                                       FAIL1020
C     UNCONFINED CONCRETE STRESSES                                      FAIL1030
      LBGN = INDEX+(L-1)*40+6                                           FAIL1040
      LEND = LBGN+35                                                    FAIL1050
      K = 0                                                             FAIL1060
      DO 3050 KK=LBGN,LEND,8                                            FAIL1070
      K = K+1                                                           FAIL1080
      STRESS(K) = DATA(KK)                                              FAIL1090
 3050 CONTINUE                                                          FAIL1100
      INDEX = KDATA(LPI+M)+NGRPM*5+(L-1)*5+20                           FAIL1110
      STRAIN(1) = JATA(INDEX+1)                                         FAIL1120
      STRAIN(5) = DATA(INDEX+5)                                         FAIL1130
C                                                                       FAIL1140
C     CONFINED CONCRETE STRESSES                                        FAIL1150
      INDEX = KDATA(LPSI+M)+NGRPM*40+(L-1)*72+319                       FAIL1160
      LBGN = INDEX+6                                                    FAIL1170
      LEND = LBGN+67                                                    FAIL1180
```

```
      DO 3060 KK=LBGN,LEND,8                                            FAIL1190
      K = K+1                                                           FAIL1200
      STRESS(K) = DATA(KK)                                              FAIL1210
 3060 CONTINUE                                                          FAIL1220
C                                                                       FAIL1230
C     REINFORCING STEEL STRAINS.                                        FAIL1240
      INDEX = KDATA(LPI+M)-1+NGRPM*3                                    FAIL1250
      LBGN = INDEX + L                                                  FAIL1260
      LEND = INDEX+NGRPM*2                                              FAIL1270
      DO 3070 KK=LBGN,LEND,2                                            FAIL1280
      K = K+1                                                           FAIL1290
      STRAIN(K) = DATA(KK)                                              FAIL1300
 3070 CONTINUE                                                          FAIL1310
      INDEX = KDATA(LPSI+M)-1+NGRPM*24+(L-1)*8                          FAIL1320
      LBGN = INDEX+5                                                    FAIL1330
      LEND = INDEX+NGRPM*16                                             FAIL1340
C                                                                       FAIL1350
C     REINFORCING STEEL STRESSES.                                       FAIL1360
      K = 14                                                            FAIL1370
      DO 3080 KK=LBGN,LEND,16                                           FAIL1380
      K = K+1                                                           FAIL1390
      SFIVE(K) = DATA(KK)                                               FAIL1400
      SEIGHT(K) = DATA(KK+3)                                            FAIL1410
 3080 CONTINUE                                                          FAIL1420
      DO 3090 K=15,NNG                                                  FAIL1430
      STRESS(K) = -SEIGHT(K)                                            FAIL1440
      IF (STRAIN(K)-SFIVE(K).GE.0.E0) STRESS(K) = SEIGHT(K)             FAIL1450
 3090 CONTINUE                                                          FAIL1460
 3100 SAVCRV(L,M) = (STRAIN(1)-STRAIN(5))/(TEMP(1)-TEMP(5))             FAIL1470
C                                                                       FAIL1480
```

```
C     STRESS RESULTANTS DUE TO UNCONFINED CONCRETE
      DO 3110 K=1,4
      WIDTH = BMEM(M)
      IF(K.EQ.2.OR.K.EQ.3) WIDTH = WIDTH - BPP(M)
      FORCE = (STRESS(K)+STRESS(K+1))/2.E0*WIDTH*(TEMP(K)-TEMP(K+1))
      IF(ABS(FORCE).LE.TINY) GO TO 3110
      IF (L.EQ.2) SAVAXL(2,M) = SAVAXL(2,M) + FORCE
      CENT=(TEMP(K)-TEMP(K+1))*(2.E0*STRESS(K)+STRESS(K+1))/
     1(STRESS(K)+STRESS(K+1))/3.E0
      SAVMOM(L,M) = SAVMOM(L,M)+FORCE*(CENT+TEMP(K+1))
 3110 CONTINUE
C
C     STRESS RESULTANTS DUE TO CONFINED CONCRETE
      DO 3130 K=6,13
      AREA = BPP(M)*(TEMP(K)-TEMP(K+1))
C
C     REMOVE CONCRETE DISPLACED BY REINFORCING STEEL
      DO 3120 N=1,NGRPM
      IF(YBAR(N,M).LT.TEMP(K).AND.YBAR(N,M).GT.TEMP(K+1))
     *                   AREA = AREA - AGRP(N,M)
      IF(YBAR(N,M).EQ.TEMP(K).OR.YBAR(N,M).EQ.TEMP(K+1))
     *                   AREA = AREA - AGRP(N,M) /2.E0
 3120 CONTINUE
      FORCE = (STRESS(K)+STRESS(K+1))*AREA/2.E0
      IF(ABS(FORCE).LE.TINY) GO TO 3130
      IF(L.EQ.2) SAVAXL(2,M) = SAVAXL(2,M) + FORCE
      CENT=(TEMP(K)-TEMP(K+1))*(2.E0*STRESS(K)+STRESS(K+1))/
     1(STRESS(K)+STRESS(K+1))/3.E0
      SAVMOM(L,M) = SAVMOM(L,M)+FORCE*(CENT+TEMP(K+1))
 3130 CONTINUE
```

```
C                                                                       FAIL1790
C     STRESS RESULTANTS DUE TO REINFORCING STEEL                        FAIL1800
      DO 3140 K=15,NNG                                                  FAIL1810
      FORCE = STRESS(K)*AGRP(K-14,M)                                    FAIL1820
      IF(L.EQ.2) SAVAXL(2,M) = SAVAXL(2,M) + FORCE                      FAIL1830
      SAVMOM(L,M) = SAVMOM(L,M)+FORCE*TEMP(K)                           FAIL1840
 3140 CONTINUE                                                          FAIL1850
      IF(L.EQ.2) SAVMOM(2,M)=-SAVMOM(2,M)                               FAIL1860
      IF(L.EQ.1) ETOPA=STRAIN(1)                                        FAIL1870
      IF(L.EQ.1) EBOTA=STRAIN(5)                                        FAIL1880
      IF(L.EQ.2) ETOPB=STRAIN(1)                                        FAIL1890
      IF(L.EQ.2) EBOTB=STRAIN(5)                                        FAIL1900
      ECMAX=AMAX1(ETOPA,EBOTA,ETOPB,EBOTB)                              FAIL1910
      ECMIN = AMIN1 (ETOPA,EBOTA,ETOPB,EBOTB)                           FAIL1920
 3150 CONTINUE                                                          FAIL1930
      SAVAXL(1,M) = -SAVAXL(2,M)                                        FAIL1940
      SAVSHR(2,M) = -(SAVMOM(1,M)+SAVMOM(2,M))/XLEN+SAVAXL(2,M)*UY      FAIL1950
      SAVSHR(1,M) = -SAVSHR(2,M)                                        FAIL1960
      GO TO 5000                                                        FAIL1970
C                                                                       FAIL1980
C     WIDE FLANGE STRESSES AND STRAINS.                                 FAIL1990
C                                                                       FAIL2000
3170  CALL WIDE (M,UR,UD,3)                                             FAIL2010
      DO 4080 L=1,2                                                     FAIL2020
      IF(MSTAT(M).EQ.3) GO TO 4020                                      FAIL2030
C                                                                       FAIL2040
C     ELASTIC STRESSES AND STRAINS                                      FAIL2050
      J = MATW(M)                                                       FAIL2060
      XLOC = 0.E0                                                       FAIL2070
      IF(L.EQ.2) XLOC = XLEN                                            FAIL2080
```

```
      DO 4010 K=1,11                                                    FAIL2090
      CALL STRN(M,XLOC,YFIBR(K,M),STRAIN(K))                            FAIL2100
      IF (STRAIN(K).LE.0.E0) STRESS(K) = STRAIN(K)*EC(J)                FAIL2110
      IF(STRAIN(K).GT.0.E0) STRESS(K) = STRAIN(K)*ET(J)                 FAIL2120
 4010 CONTINUE                                                          FAIL2130
      GO TO 4060                                                        FAIL2140
C                                                                       FAIL2150
C     INELASTIC STRESSES AND STRAINS                                    FAIL2160
 4020 INDEX = KDATA(LPI+M)+21                                           FAIL2170
      LBGN = INDEX+(L-1)*11                                             FAIL2180
      LEND = LBGN+10                                                    FAIL2190
      K = 0                                                             FAIL2200
      DO 4030 KK=LBGN,LEND                                              FAIL2210
      K = K+1                                                           FAIL2220
      STRAIN(K) = DATA(KK)                                              FAIL2230
 4030 CONTINUE                                                          FAIL2240
      INDEX = KDATA(LPSI+M)+167                                         FAIL2250
      LBGN = INDEX+(L-1)*88+5                                           FAIL2260
      LEND = LBGN + 80                                                  FAIL2270
      K = 0                                                             FAIL2280
      DO 4040 KK=LBGN,LEND,8                                            FAIL2290
      K = K+1                                                           FAIL2300
      SFIVE(K) = DATA(KK)                                               FAIL2310
      SEIGHT(K) = DATA(KK+3)                                            FAIL2320
 4040 CONTINUE                                                          FAIL2330
      DO 4050 K=1,11                                                    FAIL2340
      STRESS(K) = -SEIGHT(K)                                            FAIL2350
      IF (STRAIN(K)-SFIVE(K).GE.0.E0) STRESS(K) = SEIGHT(K)             FAIL2360
 4050 CONTINUE                                                          FAIL2370
 4060 SAVCRV(L,M) = (STRAIN(1)-STRAIN(11))/(YFIBR(1,M)-YFIBR(11,M))     FAIL2380
```

```
      DO 4070 K=1,10                                                    FAIL2390
      WIDTH = TWWF(M)                                                   FAIL2400
      IF(K.EQ.1.OR.K.EQ.10) WIDTH = BWF(M)                              FAIL2410
      FORCE=(STRESS(K)+STRESS(K+1))*WIDTH*(YFIBR(K,M)-YFIBR(K+1,M))/2.E0FAIL2420
      IF(ABS(FORCE).LE.TINY) GO TO 4070                                 FAIL2430
      IF(L.EQ.2) SAVAXL(2,M) = SAVAXL(2,M) + FORCE                      FAIL2440
      CENT=(YFIBR(K,M)-YFIBR(K+1,M))*(2.E0*STRESS(K)+STRESS(K+1))/     FAIL2450
     1(STRESS(K)+STRESS(K+1))/3.E0                                      FAIL2460
      SAVMOM(L,M) = SAVMOM(L,M)+FORCE*(CENT+YFIBR(K+1,M))               FAIL2470
 4070 CONTINUE                                                          FAIL2480
      IF(L.EQ.2) SAVMOM(2,M)=-SAVMOM(2,M)                               FAIL2490
      IF(L.EQ.1) ETOPA=STRAIN(1)                                        FAIL2500
      IF(L.EQ.1) EBOTA=STRAIN(11)                                       FAIL2510
      IF(L.EQ.2) ETOPB=STRAIN(1)                                        FAIL2520
      IF(L.EQ.2) EBOTB=STRAIN(11)                                       FAIL2530
      ESMAX=AMAX1(ETOPA,EBOTA,ETOPB,EBOTB)                              FAIL2540
      ESMIN = AMIN1 (ETOPA,EBOTA,ETOPB,EBOTB)                           FAIL2550
 4080 CONTINUE                                                          FAIL2560
      SAVAXL(1,M) = -SAVAXL(2,M)                                        FAIL2570
      SAVSHR(2,M) = -(SAVMOM(1,M)+SAVMOM(2,M))/XLEN+SAVAXL(2,M)*UY      FAIL2580
      SAVSHR(1,M) = -SAVSHR(2,M)                                        FAIL2590
C                                                                       FAIL2600
C     DETERMINATION OF CRITERIA TO BE CHECKED.                          FAIL2610
C                                                                       FAIL2620
5000  J = MBAR(NESMAX,M)                                                FAIL2630
      IF(MTYPE(M).EQ.4) J = MATW(M)                                     FAIL2640
C                                                                       FAIL2650
C     STEEL FRACTURE CRITERIA    REF. B.3.1.2                           FAIL2660
C                                                                       FAIL2670
      IF(ABS(ESMAX).LT.STN(7,J)) GO TO 5020                             FAIL2680
```

```
      LINE = LINE + 1
      IF (LINE.GT.NL) CALL PAGE
      IFAIL = 1
      PRINT 610, IP(M),IQ(M)
  610 FORMAT( 39H ***STEEL FRACTURE DETECTED IN ELEMENT ,I3,1H-,I3,
     +      39H. THE ANALYSIS IS TERMINATED (FAIL).***)
 5020 IF(MTYPE(M).EQ.4) GO TO 7000
C
C     CONCRETE CRUSHING    REF.B.3.1.1
      IF(NTIESM.GT.0) GO TO 5313
      J = MCODE(M)
      TEMPR = (3.E0*EGSIS + .002E0*FCFY(J))/(FCFY(J)-1.E3*EGSIS)
      IF(TEMPR.GT.3.5E-3) TEMPR = 3.5E-3
      EPSTAR = CA*TEMPR
      IF(EPSTAR-ABS(ECMAX).GT.0.E0) GO TO 5313
      LINE = LINE + 1
      IF (LINE.GT.NL) CALL PAGE
      IFAIL = 1
      PRINT 630, IP(M),IQ(M)
  630 FORMAT( 42H ***CONCRETE CRUSHING DETECTED IN ELEMENT ,I3,1H-,I3,
     +      39H. THE ANALYSIS IS TERMINATED (FAIL).***)
C
C     SIMULTANEOUS BAR BUCKLING AND CONCRETE CRUSHING     REF.B.3.1.3
5313  IF(NTIESM.EQ.0) GO TO 532
      J = MBAR(NESMIN,M)
      DO 100 K=1,6
      IF((ABS(ESMIN).GT.STN(K,J).AND.ABS(ESMIN).LT.STN(K+1,J))   .OR.
     *    (ABS(ESMIN).EQ.STN(K,J))) GO TO 5030
  100 CONTINUE
 5030 ETAN = SLOPE(K,J)
```

```
      FSTRS = STS(K,J)+ETAN*ABS(ESMIN)                                  FAIL2990
      CRIT = CB*PI*PI/16.E0*(BDM(NESMIN,M)/STIES(7,M))*ETAN             FAIL3000
      IF (CRIT-FSTRS .GT. 0.E0) GO TO 532                               FAIL3010
      LINE = LINE + 1                                                   FAIL3020
      IF (LINE.GT.NL) CALL PAGE                                         FAIL3030
      IF(ISTOP.EQ.1) GO TO 5031                                         FAIL3040
      IFAIL = 1                                                         FAIL3050
      PRINT 640, IP(M),IQ(M)                                            FAIL3060
640   FORMAT( 52H ***SIMULTANEOUS BAR BUCKLING AND CONCRETE CRUSHING ,  FAIL3070
     2      20HDETECTED IN ELEMENT ,I3,1H-,I3,                          FAIL3080
     +     39H. THE ANALYSIS IS TERMINATED (FAIL).***)                  FAIL3090
      GO TO 532                                                         FAIL3100
 5031 PRINT 650, IP(M),IQ(M)                                            FAIL3110
650   FORMAT( 52H ** SIMULTANEOUS BAR BUCKLING AND CONCRETE CRUSHING ,  FAIL3120
     2      20HDETECTED IN ELEMENT ,I3,1H-,I3,                          FAIL3130
     +     39H. THE ANALYSIS IS CONTINUING (FAIL). **)                  FAIL3140
C                                                                       FAIL3150
C     SHEAR-FLEXURE CHECK        CLASSIFICATION SEGMENT                 FAIL3160
C                                                                       FAIL3170
  532 PASS(1) = .FALSE.                                                 FAIL3180
      PASS(2) = .FALSE.                                                 FAIL3190
C                                                                       FAIL3200
C     SET OPTIONAL AND CONSTANT FUNCTION VALUES                         FAIL3210
      J = MCODE(M)                                                      FAIL3220
      TEMPO = SAVAXL(2,M)/BMEM(M)/SQRT(FCFY(J))                         FAIL3230
      FFOURA = TEMPO/D(M)                                               FAIL3240
      FFOURB = TEMPO/(HMEM(M)-DP(M))                                    FAIL3250
      IF(TEMPO.GE.0.E0) ALFA = 0.025E0*(4.E0-CD)                        FAIL3260
      IF(TEMPO.LT.0.E0) ALFA = 0.05E0*(2.E0+CD)                         FAIL3270
      FTWOBR = 1.E3/BMEM(M)/SQRT(FCFY(J))                               FAIL3280
```

```
      TEMPO = ABS(SAVSHR(1,M))/BMEM(M)/SQRT(FCFY(J))                    FAIL3290
      FA = TEMPO/D(M)                                                   FAIL3300
      FB = TEMPO/(HMEM(M)-DP(M))                                        FAIL3310
C                                                                       FAIL3320
C     COMPUTE STEEL AREAS ABOVE AND BELOW CENTROID OF GROSS SECTION.    FAIL3330
C                                                                       FAIL3340
      ASTOP = 0.E0                                                      FAIL3350
      ASBOT = 0.E0                                                      FAIL3360
      ASTOT = 0.E0                                                      FAIL3370
      DO 200 I=1,NGRPM                                                  FAIL3380
      ASTOT = ASTOT+AGRP(I,M)                                           FAIL3390
      IF(HTOP(M)-HMEM(M)/2.E0.GT.YBAR(I,M)) ASBOT = ASBOT+AGRP(I,M)     FAIL3400
      IF(HTOP(M)-HMEM(M)/2.E0.EQ.YBAR(I,M)) ASBOT = ASBOT+AGRP(I,M)/2.E0FAIL3410
  200 CONTINUE                                                          FAIL3420
      ASTOP = ASTOT-ASBOT                                               FAIL3430
C                                                                       FAIL3440
C     CLASSIFY BENDING STATE BY END MOMENTS - - SIX POSSIBLE CONFIGS.   FAIL3450
C                                                                       FAIL3460
      DENOM = (EBOTB-ETOPB)-(EBOTA-ETOPA)                               FAIL3470
C                                                                       FAIL3480
C     RETURN IF SHEAR-FLEXURE FAILURE IS SUBORDINATE TO OTHER CRITERIA  FAIL3490
C                                                                       FAIL3500
      IF (ABS(DENOM).LE.TINY) GO TO 7000                                FAIL3510
      XX = (ETOPA-EBOTA)/DENOM*XL(M)                                    FAIL3520
      KCH = 1                                                           FAIL3530
      IF (XX.GE.0.E0 .AND. XX.LE.XLEN) GO TO 30                         FAIL3540
      KCH = 2                                                           FAIL3550
      CHL = XLEN                                                        FAIL3560
      IF(XX.LE.0.E0.AND.EBOTB.GT.ETOPB) GO TO 11                        FAIL3570
      IF(XX.LE.0.E0.AND.EBOTB.LT.ETOPB) GO TO 21                        FAIL3580
```

```
      IF(XX.GT.CHL.AND.EBOTA.GT.ETOPA) GO TO 12                         FAIL3590
      IF(XX.GE.CHL.AND.EBOTA.LT.ETOPA) GO TO 22                         FAIL3600
   30 IF(EBOTA.LT.ETOPA) GO TO 31                                       FAIL3610
      IF(EBOTA.GT.ETOPA) GO TO 32                                       FAIL3620
C                                                                       FAIL3630
C     LARGER POSITIVE MOMENT AT RIGHT (Q) END OF ELEMENT . . . .        FAIL3640
C     . . .  ZERO MOMENT OUTSIDE SPAN LENGTH                            FAIL3650
C                                                                       FAIL3660
   11 DCHK = D(M)                                                       FAIL3670
      DFAC = CHL/DCHK/2.E0-0.5E0                                        FAIL3680
      IF(DFAC.LT.0.5E0) DFAC = 0.5E0                                    FAIL3690
      IF(DFAC.GT.1.5E0) DFAC = 1.5E0                                    FAIL3700
      EMHD = DFAC*DCHK                                                  FAIL3710
C                                                                       FAIL3720
C     RETURN IF POTENTIAL CRACK IS WITHIN JOINT                         FAIL3730
C                                                                       FAIL3740
      IF(EMHD.LE.(CHL-EFLM(M)-XBEGM(M))) GO TO 7000                     FAIL3750
      XCS = CHL-XX-EMHD                                                 FAIL3760
      FFOUR(2) = FFOURA                                                 FAIL3770
      ASUBS = ASBOT                                                     FAIL3780
      FTWO(2) = FTWOBR*ASUBS/XCS                                        FAIL3790
      F(2) = FA                                                         FAIL3800
      DO 111 N=1,NTIESM                                                 FAIL3810
      IF(XBEGS(N,M).LE.XL(M)-EMHD.AND.XL(M)-EMHD.LE.XBEGS(N,M)+         FAIL3820
     *   NSPAC(N,M)*STIES(N,M))  ASUBV(2) = ATIES(N,M)                  FAIL3830
  111 CONTINUE                                                          FAIL3840
      GO TO 5321                                                        FAIL3850
C                                                                       FAIL3860
C     LARGER POSITIVE MOMENT AT LEFT (P) END OF ELEMENT . . . .         FAIL3870
C     . . .  ZERO MOMENT OUTSIDE SPAN LENGTH                            FAIL3880
```

```
C                                                                       FAIL3890
   12 DCHK = D(M)                                                       FAIL3900
      DFAC = CHL/DCHK/2.E0-0.5E0                                        FAIL3910
      IF(DFAC.LT.0.5E0) DFAC = 0.5E0                                    FAIL3920
      IF(DFAC.GT.1.5E0) DFAC = 1.5E0                                    FAIL3930
      EMHD = DFAC*DCHK                                                  FAIL3940
C                                                                       FAIL3950
C     RETURN IF POTENTIAL CRACK IS WITHIN JOINT                         FAIL3960
C                                                                       FAIL3970
      IF(EMHD.LE.XBEGM(M)) GO TO 7000                                   FAIL3980
      XCS = XX-EMHD                                                     FAIL3990
      FFOUR(2) = FFOURA                                                 FAIL4000
      ASUBS = ASBOT                                                     FAIL4010
      FTWO(2) = FTWOBR*ASUBS/XCS                                        FAIL4020
      F(2) = FA                                                         FAIL4030
      DO 122 N=1,NTIESM                                                 FAIL4040
      IF(XBEGS(N,M).LE.EMHD.AND.EMHD.LE.XBEGS(N,M)+NSPAC(N,M)*          FAIL4050
     *   STIES(N,M))  ASUBV(2) = ATIES(N,M)                             FAIL4060
  122 CONTINUE                                                          FAIL4070
      GO TO 5321                                                        FAIL4080
C                                                                       FAIL4090
C     LARGER NEGATIVE MOMENT AT RIGHT (Q) END OF ELEMENT . . . . .      FAIL4100
C     . . . .  ZERO MOMENT OUTSIDE SPAN LENGTH                          FAIL4110
C                                                                       FAIL4120
   21 DCHK = HMEM(M)-DP(M)                                              FAIL4130
      DFAC = CHL/DCHK/2.E0-0.5E0                                        FAIL4140
      IF(DFAC.LT.0.5E0) DFAC = 0.5E0                                    FAIL4150
      IF(DFAC.GT.1.5E0) DFAC = 1.5E0                                    FAIL4160
      EMHD = DFAC*DCHK                                                  FAIL4170
C                                                                       FAIL4180
```

```
C     RETURN IF POTENTIAL CRACK IS WITHIN JOINT                         FAIL4190
C                                                                       FAIL4200
      IF(EMHD.LE.(CHL-EFLM(M)-XBEGM(M))) GO TO 7000                     FAIL4210
      XCS = CHL-XX-EMHD                                                 FAIL4220
      FFOUR(2) = FFOURB                                                 FAIL4230
      ASUBS = ASTOP                                                     FAIL4240
      FTWO(2) = FTWOBR*ASUBS/XCS                                        FAIL4250
      F(2) = FB                                                         FAIL4260
      DO 211 N=1,NTIESM                                                 FAIL4270
      IF(XBEGS(N,M).LE.XL(M)-EMHD.AND.XL(M)-EMHD.LE.XBEGS(N,M)+         FAIL4280
     *   NSPAC(N,M)*STIES(N,M))  ASUBV(2) = ATIES(N,M)                  FAIL4290
  211 CONTINUE                                                          FAIL4300
      GO TO 5321                                                        FAIL4310
C                                                                       FAIL4320
C     LARGER NEGATIVE MOMENT AT LEFT(P) END OF ELEMENT . . . .          FAIL4330
C     . . .    ZERO MOMENT OUTSIDE SPAN LENGTH                          FAIL4340
C                                                                       FAIL4350
   22 DCHK = HMEM(M)-DP(M)                                              FAIL4360
      DFAC = CHL/DCHK/2.E0-0.5E0                                        FAIL4370
      IF(DFAC.LT.0.5E0) DFAC = 0.5E0                                    FAIL4380
      IF(DFAC.GT.1.5E0) DFAC = 1.5E0                                    FAIL4390
      EMHD = DFAC*DCHK                                                  FAIL4400
C                                                                       FAIL4410
C     RETURN IF POTENTIAL CRACK IS WITHIN JOINT                         FAIL4420
C                                                                       FAIL4430
      IF(EMHD.LE.XBEGM(M)) GO TO 7000                                   FAIL4440
      XCS = XX-EMHD                                                     FAIL4450
      FFOUR(2) = FFOURB                                                 FAIL4460
      ASUBS = ASTOP                                                     FAIL4470
      FTWO(2) = FTWOBR*ASUBS/XCS                                        FAIL4480
```

```
      F(2) = FB                                                         FAIL4490
      DO 222 N=1,NTIESM                                                 FAIL4500
      IF(XBEGS(N,M).LE.EMHD.AND.EMHD.LE.XBEGS(N,M)+NSPAC(N,M)*          FAIL4510
     *   STIES(N,M))  ASUBV(2) = ATIES(N,M)                             FAIL4520
  222 CONTINUE                                                          FAIL4530
      GO TO 5321                                                        FAIL4540
C                                                                       FAIL4550
C     POSITIVE MOMENT AT RIGHT (Q) END OF ELEMENT . . . .               FAIL4560
C     . . .   ZERO MOMENT WITHIN SPAN LENGTH                            FAIL4570
C                                                                       FAIL4580
C     SET FUNCTION VALUES FOR LEFT PORTION (NEGATIVE MOMENT)            FAIL4590
C                                                                       FAIL4600
   31 DCHK = HMEM(M)-DP(M)                                              FAIL4610
      CHL = XX                                                          FAIL4620
      DFAC = CHL/DCHK/2.E0-0.5E0                                        FAIL4630
      IF(DFAC.LT.0.5E0) DFAC = 0.5E0                                    FAIL4640
      IF(DFAC.GT.1.5E0) DFAC = 1.5E0                                    FAIL4650
      EMHD = DFAC*DCHK                                                  FAIL4660
C                                                                       FAIL4670
C     SET BYPASS FLAG FOR THIS PORTION IF POTENTIAL CRACK IS WITHIN JOINFAIL4680
C                                                                       FAIL4690
      IF(EMHD.LE.XBEGM(M)) PASS(1) = .TRUE.                             FAIL4700
      XCS = CHL-EMHD                                                    FAIL4710
      FFOUR(1) = FFOURB                                                 FAIL4720
      ASUBS = ASTOP                                                     FAIL4730
      FTWO(1) = FTWOBR*ASUBS/XCS                                        FAIL4740
      F(1) = FB                                                         FAIL4750
      DO 311 N=1,NTIESM                                                 FAIL4760
      IF(XBEGS(N,M).LE.EMHD.AND.EMHD.LE.XBEGS(N,M)+NSPAC(N,M)*          FAIL4770
     *   STIES(N,M))  ASUBV(1) = ATIES(N,M)                             FAIL4780
```

```
  311 CONTINUE                                                          FAIL4790
C                                                                       FAIL4800
C     SET FUNCTION VALUES FOR RIGHT PORTION (POSITIVE MOMENT)           FAIL4810
C                                                                       FAIL4820
      DCHK = D(M)                                                       FAIL4830
      CHL = XL(M)-XX                                                    FAIL4840
      DFAC = CHL/DCHK/2.E0-0.5E0                                        FAIL4850
      IF(DFAC.LT.0.5E0) DFAC = 0.5E0                                    FAIL4860
      IF(DFAC.GT.1.5E0) DFAC = 1.5E0                                    FAIL4870
      EMHD = DFAC*DCHK                                                  FAIL4880
C                                                                       FAIL4890
C     SET BYPASS FLAG FOR THIS PORTION IF POTENTIAL CRACK IS WITHIN JOINFAIL4900
C                                                                       FAIL4910
      IF(EMHD.LE.XL(M)-EFLM(M)-XBEGM(M)) PASS(2) = .TRUE.               FAIL4920
C                                                                       FAIL4930
C     RETURN IF BOTH BYPASS FLAGS HAVE BEEN SET                         FAIL4940
C                                                                       FAIL4950
      IF(PASS(1).AND.PASS(2)) GO TO 7000                                FAIL4960
      XCS = CHL-EMHD                                                    FAIL4970
      FFOUR(2) = FFOURA                                                 FAIL4980
      ASUBS = ASBOT                                                     FAIL4990
      FTWO(2) = FTWOBR*ASUBS/XCS                                        FAIL5000
      F(2) = FA                                                         FAIL5010
      DO 312 N=1,NTIESM                                                 FAIL5020
      IF(XBEGS(N,M).LE.XL(M)-EMHD.AND.XL(M)-EMHD.LE.XBEGS(N,M)+         FAIL5030
     *   NSPAC(N,M)*STIES(N,M))  ASUBV(2) = ATIES(N,M)                  FAIL5040
  312 CONTINUE                                                          FAIL5050
      GO TO 5321                                                        FAIL5060
                                                                        FAIL5070
C     POSITIVE MOMENT AT LEFT (P) END OF ELEMENT . . . .                FAIL5080
```

```
C     . . .   ZERO MOMENT WITHIN SPAN LENGTH                            FAIL5090
C                                                                       FAIL5100
C     SET FUNCTION VALUES FOR LEFT PORTION (POSITIVE MOMENT)            FAIL5110
C                                                                       FAIL5120
   32 DCHK = D(M)                                                       FAIL5130
      CHL = XX                                                          FAIL5140
      DFAC = CHL/DCHK/2.E0-0.5E0                                        FAIL5150
      IF(DFAC.LT.0.5E0) DFAC = 0.5E0                                    FAIL5160
      IF(DFAC.GT.1.5E0) DFAC = 1.5E0                                    FAIL5170
      EMHD = DFAC*DCHK                                                  FAIL5180
C                                                                       FAIL5190
C     SET BYPASS FLAG FOR THIS PORTION IF POTENTIAL CRACK IS WITHIN JOINFAIL5200
C                                                                       FAIL5210
      IF(EMHD.LE.XBEGM(M)) PASS(1) = .TRUE.                             FAIL5220
      XCS = CHL-EMHD                                                    FAIL5230
      FFOUR(1) = FFOURA                                                 FAIL5240
      ASUBS = ASBOT                                                     FAIL5250
      FTWO(1) = FTWOBR*ASUBS/XCS                                        FAIL5260
      F(1) = FA                                                         FAIL5270
      DO 321 N=1,NTIESM                                                 FAIL5280
      IF(XBEGS(N,M).LE.EMHD.AND.EMHD.LE.XBEGS(N,M)+NSPAC(N,M)*          FAIL5290
     *   STIES(N,M))  ASUBV(1) = ATIES(N,M)                             FAIL5300
  321 CONTINUE                                                          FAIL5310
C                                                                       FAIL5320
C     SET FUNCTION VALUES FOR RIGHT PORTION (NEGATIVE MOMENT)           FAIL5330
C                                                                       FAIL5340
      DCHK = HMEM(M)-DP(M)                                              FAIL5350
      CHL = XL(M)-XX                                                    FAIL5360
      DFAC = CHL/DCHK/2.E0-0.5E0                                        FAIL5370
      IF(DFAC.LT.0.5E0) DFAC = 0.5E0                                    FAIL5380
```

```
      IF(DFAC.GT.1.5E0) DFAC = 1.5E0                                    FAIL5390
      EMHD = DFAC*DCHK                                                  FAIL5400
C                                                                       FAIL5410
C     SET BYPASS FLAG FOR THIS PORTION IF POTENTIAL CRACK IS WITHIN JOINFAIL5420
C                                                                       FAIL5430
      IF(EMHD.LE.XL(M)-EFLM(M)-XBEGM(M)) PASS(2) = .TRUE.               FAIL5440
C                                                                       FAIL5450
C     RETURN IF BOTH BYPASS FLAGS HAVE BEEN SET                         FAIL5460
C                                                                       FAIL5470
      IF(PASS(1).AND.PASS(2)) GO TO 7000                                FAIL5480
      XCS = CHL-EMHD                                                    FAIL5490
      FFOUR(2) = FFOURB                                                 FAIL5500
      ASUBS = ASTOP                                                     FAIL5510
      FTWO(2) = FTWOBR*ASUBS/XCS                                        FAIL5520
      F(2) = FB                                                         FAIL5530
      DO 322 N=1,NTIESM                                                 FAIL5540
      IF(XBEGS(N,M).LE.XL(M)-EMHD.AND.XL(M)-EMHD.LE.XBEGS(N,M)+         FAIL5550
     *   NSPAC(N,M)*STIES(N,M))  ASUBV(2) = ATIES(N,M)                  FAIL5560
  322 CONTINUE                                                          FAIL5570
C                                                                       FAIL5580
C     SHEAR-FLEXURE CRITERIA                                            FAIL5590
C                                                                       FAIL5600
 5321 IF(ECMIN*ECMAX.GE.0.E0) GO TO 7000                                FAIL5610
      DO 400 N=KCH,2                                                    FAIL5620
      IF(PASS(N)) GO TO 400                                             FAIL5630
      TEMPO = 1.5E0+CC*FTWO(N)                                          FAIL5640
      TCHK = 3.E0*(7.E0+2.E0*CC)/14.E0                                  FAIL5650
      IF(TEMPO.GT.TCHK) TEMPO = TCHK                                    FAIL5660
      FSTAR = TEMPO-ALFA*FFOUR(N)                                       FAIL5670
C                                                                       FAIL5680
```

```
C     DETECTION OF THE PRINCIPAL DIAGONAL CRACK LEADING TO FAILURE(3.2.1FAIL5690
C                                                                       FAIL5700
      IF(NTIESM.GT.0) GO TO 5050                                        FAIL5710
      IF(FSTAR-F(N).GT.0.E0) GO TO 400                                  FAIL5720
      IF(ISTOP.EQ.1) GO TO 5040                                         FAIL5730
      LINE = LINE + 1                                                   FAIL5740
      IF (LINE.GT.NL) CALL PAGE                                         FAIL5750
      IFAIL = 1                                                         FAIL5760
      PRINT 660, IP(M),IQ(M)                                            FAIL5770
  660 FORMAT( 49H ***PRINCIPAL DIAGONAL CRACK DETECTED IN ELEMENT ,I3,  FAIL5780
     2     1H-,I3,39H. THE ANALYSIS IS TERMINATED (FAIL).***)           FAIL5790
      GO TO 400                                                         FAIL5800
 5040 PRINT 670, IP(M),IQ(M)                                            FAIL5810
  670 FORMAT( 49H ** PRINCIPAL DIAGONAL CRACK DETECTED IN ELEMENT ,I3,  FAIL5820
     2     1H-,I3,39H. THE ANALYSIS IS CONTINUING (FAIL). **)           FAIL5830
      GO TO 400                                                         FAIL5840
C                                                                       FAIL5850
C     PRINCIPAL DIAGONAL CRACK PLUS YIELDING OF WEB REINFORCEMENT(3.2.2)FAIL5860
C                                                                       FAIL5870
5050  K = MTIES(M)                                                      FAIL5880
      TEMPO = CE*STIES(7,M)*HMEM(M)/ASUBV(N)*SQRT(FCFY(J))/FCFY(K)*     FAIL5890
     *     (F(N)-FSTAR)                                                 FAIL5900
      IF(1.E0-TEMPO.GT.0.E0) GO TO 400                                  FAIL5910
      IF(ISTOP.EQ.1) GO TO 5060                                         FAIL5920
      LINE = LINE + 1                                                   FAIL5930
      IF (LINE.GT.NL) CALL PAGE                                         FAIL5940
      IFAIL = 1                                                         FAIL5950
      PRINT 680, IP(M),IQ(M)                                            FAIL5960
  680 FORMAT( 53H ***PRINCIPAL DIAGONAL CRACK PLUS WEB STEEL YIELDING , FAIL5970
     2      20HDETECTED IN ELEMENT ,I3,1H-,I3,                          FAIL5980
```

```
CFORK    0  10
      SUBROUTINE FORK (J1,I)                                            FORK   0
C                                                                       FORK  10
C     SUBROUTINE TO INTERPRET TYPE OF DATA CARD IN BEAM.                FORK  20
C                                                                       FORK  30
C THE VALUE OF I IS SET TO INDICATE THE TYPE OF CARD.                   FORK  40
C I = 1, ELEMENT PARAMETER CARD.                                        FORK  50
C I = 2, CONCRETE DATA CARD.                                            FORK  60
C I = 3, LONGITUDINAL  REINFORCEMENT DATA CARD.                         FORK  70
C I = 4, STIRRUP DATA CARD.                                             FORK  80
C I = 5, TIE DATA CARD.                                                 FORK  90
C I = 6, WIDE FLANGE DATA CARD.                                         FORK 100
C I = 7, LEAF SPRING DATA CARD.                                         FORK 110
C I = 8, DATA CARD CANNOT BE INTERPRETED.                               FORK 120
C I = 9, END CARD OF ELEMENT DATA BLOCK.                                FORK 130
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),       FORK 140
     1   MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),FORK 150
     2   MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)                       FORK 160
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,FORK 170
     1   IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,FORK 180
     2   NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,  FORK 190
     3   NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTA3,NTAPE,   FORK 200
     4   NTIMES,NVEL,IINITD                                             FORK 210
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),FORK 220
     1   BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),     FORK 230
     2   HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),FORK 240
     3   TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                 FORK 250
     4   XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45),  FORK 260
     5  YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)                FORK 270
      DATA IA,IB,IC,ID,IE,IF/4H    ,4HBARS,4HSTIR,4HTIES,4HWFST,4HLEAF/ FORK 280
```

```
CFITS    0  10
      SUBROUTINE FITS (STEP,FVAL,DSTEP,AVAL,I)                          FITS   0
C                                                                       FITS  10
C     THIS SUBROUTINE FITS A PARABOLIC CURVE TO OBTAIN A BETTER ESTIMATEFITS  20
C     OF THE LOCATION OF THE MINIMUM.                                   FITS  30
C                                                                       FITS  40
      DIMENSION STEP(4),FVAL(4),DSTEP(4)                                FITS  50
C                                                                       FITS  60
      DSTEP(1)= STEP(I+1) - STEP(I+2)                                   FITS  70
      DSTEP(3)=STEP(I+3)-STEP(I+2)                                      FITS  80
      DF1=DSTEP(1)*(FVAL(I+3)-FVAL(I+2))                                FITS  90
      DF3=DSTEP(3)*(FVAL(I+1)-FVAL(I+2))                                FITS 100
      DF=DF1-DF3                                                        FITS 110
      IF (DF.EQ.0.E0)  GO TO 20                                         FITS 120
      DSTEP(2)=.5E0*(DSTEP(1)*DF1-DSTEP(3)*DF3)/(DF1-DF3)               FITS 130
      AVAL=(DF3-DF1)*SIGN(1.E0,DSTEP(1))*SIGN(1.E0,DSTEP(3))*SIGN(1.E0, FITS 140
     1 DSTEP(1)-DSTEP(3))                                               FITS 150
      GO TO 30                                                          FITS 160
20    DSTEP(2)=.5E0*(DSTEP(1)+DSTEP(3))                                 FITS 170
      AVAL=0.E0                                                         FITS 180
30    RETURN                                                            FITS 190
      END                                                               FITS 200
```

```
     +      39H. THE ANALYSIS IS TERMINATED (FAIL).***)                    FAIL5990
      GO TO 400                                                            FAIL6000
 5060 PRINT 690, IP(M),IQ(M)                                               FAIL6010
  690 FORMAT( 53H ** PRINCIPAL DIAGONAL CRACK PLUS WEB STEEL YIELDING ,    FAIL6020
     2      20HDETECTED IN ELEMENT ,I3,1H-,I3,                             FAIL6030
     +      39H. THE ANALYSIS IS CONTINUING (FAIL). **)                    FAIL6040
  400 CONTINUE                                                             FAIL6050
 7000 CONTINUE                                                             FAIL6060
      RETURN                                                               FAIL6070
      END                                                                  FAIL6080
```

```
      DATA IG/4H0000/
      I=8
      IF (J1.EQ.IA) I=1
      IF (J1.EQ.IB) I=3
      IF (J1.EQ.IC) I=4
      IF (J1.EQ.ID) I=5
      IF (J1.EQ.IE) I=6
      IF (J1.EQ.IF) I=7
      IF (J1.EQ.IG) I=9
      IF (I.NE.8) GO TO 30
      CALL MATY (J1,MATN)
      IF (MATN.NE.0) GO TO 20
      IF (LINE.GT.NL) CALL PAGE
      IERR=IERR+1
      LINE=LINE+1
   10 FORMAT (1H ,30H***FIRST WORD OF ELEMENT CARD ,I3,1H,,A4,60H IS NEI
     1THER A COMPONENT TYPE NOR A MATERIAL NAME (FORK)***  )
      PRINT 10, ICARD,J1
      GO TO 30
   20 I=2
   30 RETURN
      END
```

```
CFORS    0  10
      SUBROUTINE FORS (T)
C
C     THIS SUBROUTINE EVALUATES THE JOINT FORCING FUNCTIONS AT A SPECIFI
C
      COMMON DATA(10000),KDATA(500)
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50),
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),
     2    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20),
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,
     4    NTIMES,NVEL
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,
     1    LTABI,NMAX,NMAXI
      INTEGER HEAD,DHEAD
C
C     INITIALIZE STORAGE LOCATIONS
C
      DO 20 K=1,3
      DO 20 J=1,NJD
20    F(K,J)=0.0E0
      DO 90 I=1,NFF
C
C     DETERMINE DATA STORAGE LOCATIONS
C
      LLL=LFF+4*(I-1)
```

```
      LLI=LFFI+5*(I-1)                                                  FORS 290
C                                                                       FORS 300
C     RETRIEVE FORCING FUNCTION DATA                                    FORS 310
C                                                                       FORS 320
C     JL   JOINT NUMBER (LOADED)                                        FORS 330
C     JR   JOINT NUMBER (REFERENCE)                                     FORS 340
C     KK   DIRECTION (0=NORMAL TO ELEMENT,1=X,2=Y,3=Z)                  FORS 350
C     LL   REFERENCE FUNCTION NUMBER (GT.0=TABLE,-1=CONST,-2=SINE,-3=COS)FORS 360
C     MM   NUMBER OF FORWARD TIME POINT USED IN LAST EVALUATION         FORS 370
C     A    SCALE FACTOR ,I.E.,FUNCTION=VALUE*(A)                        FORS 380
C     B    MAGNITUDE SHIFT (ALSO MAGNITUDE OF CONSTANT FORCE)           FORS 390
C     C    TIME SHIFT                                                   FORS 400
C     D    TIME PERIOD FOR SINE AND COSINE REFERENCE FUNCTIONS          FORS 410
C                                                                       FORS 420
      JL=KDATA(LLI+1)                                                   FORS 430
      JR=KDATA(LLI+2)                                                   FORS 440
      KK=KDATA(LLI+3)-1                                                 FORS 450
      LL=KDATA(LLI+4)                                                   FORS 460
      MM=KDATA(LLI+5)                                                   FORS 470
      A=DATA(LLL+1)                                                     FORS 480
      B=DATA(LLL+2)                                                     FORS 490
      C=DATA(LLL+3)                                                     FORS 500
      D=DATA(LLL+4)                                                     FORS 510
C                                                                       FORS 520
C     CHECK IF REFERENCE FUNCTION IS A CONSTANT                         FORS 530
C                                                                       FORS 540
      IF (LL.NE.-1) GO TO 40                                            FORS 550
      VALUE=0.E0                                                        FORS 560
      GO TO 70                                                          FORS 570
40    TX=T-C                                                            FORS 580
```

```
C
C     CHECK IF REFERENCE FUNCTION IS SINE FUNCTION
C
      IF (LL.EQ.-2) GO TO 50
C
C     CHECK IF REFERENCE FUNCTION IS COSINE FUNCTION
C
      IF (LL.EQ.-3) GO TO 60
C
C     EVALUATE TABLED FUNCTION
C
      CALL TABL (LL,TX,MM,VALUE)
      KDATA(LLI+5)=MM
      GO TO 70
50    VALUE=SIN(2.E0*PI*TX/D)
      GO TO 70
60    VALUE=COS(2.E0*PI*TX/D)
C
C     CHECK IF DISTRIBUTED FORCES ARE APPLIED.
C
70    IF (KK.EQ.3) GO TO 80
      VALUE=VALUE*AVGL
      IF (KK.NE.0) GO TO 80
      DX=X(JR)-X(JL)
      DY=Y(JR)-Y(JL)
      XLM=SQRT(DX*DX+DY*DY)
      W=0.5E0*(VALUE*A+B)*XLM
      FJL=0.7E0*W
      FJR=0.3E0*W
      F(1,JL)=F(1,JL)-FJL*DY/XLM
```

```
      F(2,JL)=F(2,JL)+FJL*DX/XLM                                        FORS 890
      F(3,JL)=F(3,JL)+W*XLM/10.E0                                       FORS 900
      F(1,JR)=F(1,JR)-FJR*DY/XLM                                        FORS 910
      F(2,JR)=F(2,JR)+FJR*DX/XLM                                        FORS 920
      F(3,JR)=F(3,JR)-W*XLM/15.E0                                       FORS 930
      GO TO 90                                                          FORS 940
C                                                                       FORS 950
C     STORE VALUE OF JOINT FORCING FUNCTION                             FORS 960
C                                                                       FORS 970
80    F(KK,JL)=F(KK,JL)+VALUE*A+B                                       FORS 980
90    CONTINUE                                                          FORS 990
      RETURN                                                            FORS1000
      END                                                               FORS1010
```

```
CFTAB    0  10
      SUBROUTINE FTAB                                                   FTAB   0
C                                                                       FTAB  10
C     THIS SUBROUTINE READS FROM CARDS AND ERROR CHECKS DATA INPUT TO THFTAB  20
C     FUNCTION TABLES DATA BLOCK                                        FTAB  30
C                                                                       FTAB  40
      COMMON DATA(10000),KDATA(500)                                     FTAB  50
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), FTAB  60
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      FTAB  70
      INTEGER HEAD,DHEAD                                                FTAB  80
      COMMON/MAINBK/IBODY,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,FTAB  90
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,FTAB 100
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, FTAB 110
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  FTAB 120
     4    NTIMES,NVEL,IINITD                                            FTAB 130
      COMMON/SCALE/EGSIF,EGSIL                                          FTAB 140
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,          FTAB 150
     1    LTABI,NMAX,NMAXI                                              FTAB 160
C                                                                       FTAB 170
      DIMENSION ADATA(6),BDATA(6),DP(6)                                 FTAB 180
C                                                                       FTAB 190
C     INITIALIZE VARIABLES USED IN THIS SUBROUTINE                      FTAB 200
C                                                                       FTAB 210
      I=0                                                               FTAB 220
      J=0                                                               FTAB 230
      JERR=0                                                            FTAB 240
      KERR=0                                                            FTAB 250
      IIUNIT=IUNITS                                                     FTAB 260
      IIERR=IERR                                                        FTAB 270
      NTAB=0                                                            FTAB 280
```

```
      LTAB=LCURV+NDF*(NDF+1)/2
      LFF=LTAB
      LMAXI=LFFI+2
      XEGSIF = 1.E0/EGSIF
C
C     CHECK FOR AVAILABILITY OF STORAGE TO STORE FIRST SET OF DATA
      IF (LMAXI.GT.NMAXI) GO TO 170
C
C     READ BLOCK 7 TITLE CARD.
      READ (NCRD,10) DHEAD
10    FORMAT (20A4)
C
C
C     READ BLOCK 7, CARD 2 (NONE REQUIRED).
C
20    FORMAT (I4,6X,6E10.0)
      READ (NCRD,20) IA,(ADATA(M),M=1,6)
C
C     CHECK FOR NO INPUT DATA TO DATA BLOCK
      IF (IA.EQ.0) GO TO 610
30    II=I+1
C
C     CHECK SEQUENCE OF DATA TABLE NUMBERS
      IF (IA.EQ.II) GO TO 40
      IERR=IERR+1
      JERR=1
      GO TO 200
C
C     READ BLOCK 7, CARD 3 TYPE
40    READ (NCRD,20) IB,(BDATA(M),M=1,6)
```

```
      K=0                                                               FTAB 590
C                                                                       FTAB 600
C     CHECK TO SEE IF LAST DATA CARD READ IS A CONTINUATION OF SAME FUNCFTAB 610
      IF (IA.EQ.IB) GO TO 60                                            FTAB 620
      K=1                                                               FTAB 630
C                                                                       FTAB 640
C     COUNT DATA POINTS ON CARD                                         FTAB 650
      DO 50 M=3,5,2                                                     FTAB 660
      IF (ADATA(M).EQ.0.E0.AND.ADATA(M+1).EQ.0.E0) GO TO 70             FTAB 670
50    K=K+1                                                             FTAB 680
      GO TO 70                                                          FTAB 690
60    K=3                                                               FTAB 700
70    KK=2*K                                                            FTAB 710
      LMAX=LFF+KK                                                       FTAB 720
C                                                                       FTAB 730
C     CHECK FOR AVAILABLE STORAGE                                       FTAB 740
      IF (LMAX.LE.NMAX) GO TO 110                                       FTAB 750
      IERR=IERR+1                                                       FTAB 760
      KERR=1                                                            FTAB 770
      IF (IB.EQ.0) GO TO 100                                            FTAB 780
C                                                                       FTAB 790
C     IF STORAGE NOT AVAILABLE, READ REMAINING CARDS IN DATA BLOCK      FTAB 800
80    FORMAT (I5)                                                       FTAB 810
90    READ (NCRD,80) IB                                                 FTAB 820
      IF (IB.NE.0) GO TO 90                                             FTAB 830
100   IF (J.GT.0) GO TO 150                                             FTAB 840
      GO TO 200                                                         FTAB 850
110   J=J+K                                                             FTAB 860
C                                                                       FTAB 870
C     MAKE SI,ENGLISH CONVERSIONS.                                      FTAB 880
```

```
      DO 120 M=2,6,2                                                    FTAB 890
  120 ADATA(M) = ADATA(M)*EGSIF                                         FTAB 900
C                                                                       FTAB 910
C     STORE DATA                                                        FTAB 920
      DO 620 M=1,KK                                                     FTAB 930
620   DATA(LFF+M)=ADATA(M)                                              FTAB 940
      LFF=LFF+KK                                                        FTAB 950
C                                                                       FTAB 960
C     CHECK FOR NEW FUNCTION TABLE                                      FTAB 970
      IF (IA.NE.IB) GO TO 150                                           FTAB 980
C                                                                       FTAB 990
C     TRANSFER DATA FROM LAST DATA CARD READ                            FTAB1000
      DO 140 M=1,6                                                      FTAB1010
140   ADATA(M)=BDATA(M)                                                 FTAB1020
C                                                                       FTAB1030
C     RETURN TO READ NEXT DATA CARD                                     FTAB1040
      GO TO 30                                                          FTAB1050
C                                                                       FTAB1060
C     INCREMENT FUNCTION TABLE COUNTER                                  FTAB1070
150   I=I+1                                                             FTAB1080
      KDATA(LFFI+1)=LFF-2*J+1                                           FTAB1090
      KDATA(LFFI+2)=J                                                   FTAB1100
      LFFI=LFFI+2                                                       FTAB1110
      IF (IB.EQ.0) GO TO 200                                            FTAB1120
      J=0                                                               FTAB1130
C                                                                       FTAB1140
C     TRANSFER DATA FROM LAST DATA CARD READ                            FTAB1150
      IA=IB                                                             FTAB1160
      DO 160 M=1,6                                                      FTAB1170
160   ADATA(M)=BDATA(M)                                                 FTAB1180
```

```
      LMAXI=LFFI+2                                                      FTAB1190
C                                                                       FTAB1200
C     CHECK FOR AVAILABILITY OF STORAGE                                 FTAB1210
      IF (LMAXI.LE.NMAXI) GO TO 30                                      FTAB1220
170   IERR=IERR+1                                                       FTAB1230
      LERR=1                                                            FTAB1240
      IF (IB.EQ.0) GO TO 200                                            FTAB1250
C                                                                       FTAB1260
C     IF STORAGE NOT AVAILABLE, READ REMAINING CARDS IN DATA BLOCK      FTAB1270
180   READ (NCRD,190) IB                                                FTAB1280
190   FORMAT (I5)                                                       FTAB1290
      IF (IB.NE.0) GO TO 180                                            FTAB1300
C                                                                       FTAB1310
C     SET NUMBER OF FUNCTION TABLES INPUT                               FTAB1320
200   NTAB=I                                                            FTAB1330
C                                                                       FTAB1340
C     INITIALIZE OUTPUT COUNTER                                         FTAB1350
210   I=0                                                               FTAB1360
C                                                                       FTAB1370
C     IF NO OUTPUT REQUIRED, SKIP DATA OUTPUT SECTION                   FTAB1380
      IF (IPRINT.EQ.0) GO TO 240                                        FTAB1390
C                                                                       FTAB1400
C     PRINT PROBLEM DESCRIPTION AND PAGE NUMBER                         FTAB1410
C     PRINT TABLE HEADING                                               FTAB1420
      CALL PAGE                                                         FTAB1430
220   FORMAT (1H ,20A4)                                                 FTAB1440
      WRITE (NPRT,220) DHEAD                                            FTAB1450
230   FORMAT (1H0,50X,15HFUNCTION TABLES/)                              FTAB1460
      WRITE (NPRT,230)                                                  FTAB1470
      LINE=LINE+2                                                       FTAB1480
```

```
C                                                                       FTAB1490
C     CHECK FOR NO DATA STORED                                          FTAB1500
240   IF (NTAB.EQ.0) GO TO 550                                          FTAB1510
C                                                                       FTAB1520
C     INCREMENT OUTPUT COUNTER                                          FTAB1530
250   I=I+1                                                             FTAB1540
      K=KDATA(LTABI+2*I-1)                                              FTAB1550
      J=KDATA(LTABI+2*I)                                                FTAB1560
C                                                                       FTAB1570
C     IF NO OUTPUT REQUIRED, SKIP DATA OUTPUT SECTION                   FTAB1580
      IF (IPRINT.EQ.0) GO TO 310                                        FTAB1590
      LINE = LINE + 5                                                   FTAB1600
      IF (LINE.LT.NL) GO TO 270                                         FTAB1610
      CALL PAGE                                                         FTAB1620
      WRITE (NPRT,230)                                                  FTAB1630
      LINE=LINE+2                                                       FTAB1640
C                                                                       FTAB1650
C     OUTPUT HEADINGS                                                   FTAB1660
260   FORMAT(1H0,9X,18HFUNCTION TABLE NO.,I5,10X,16HNUMBER OF POINTS,I5)FTAB1670
270   WRITE (NPRT,260) I,J                                              FTAB1680
C                                                                       FTAB1690
C     OUTPUT METRIC UNITS                                               FTAB1700
280   FORMAT (/,1H ,3(7X,8HTIME (S),8X,11HFORCE (N)  ),/)               FTAB1710
      IF(IIUNIT.GE.2) WRITE (NPRT,280)                                  FTAB1720
C                                                                       FTAB1730
C     OUTPUT ENGLISH UNITS                                              FTAB1740
290   FORMAT (/,1H ,3(7X,8HTIME (S),8X,11HFORCE (LB) ),/)               FTAB1750
      IF(IIUNIT.LE.1) WRITE (NPRT,290)                                  FTAB1760
C                                                                       FTAB1770
C     PREPARE DATA FOR OUTPUT                                           FTAB1780
```

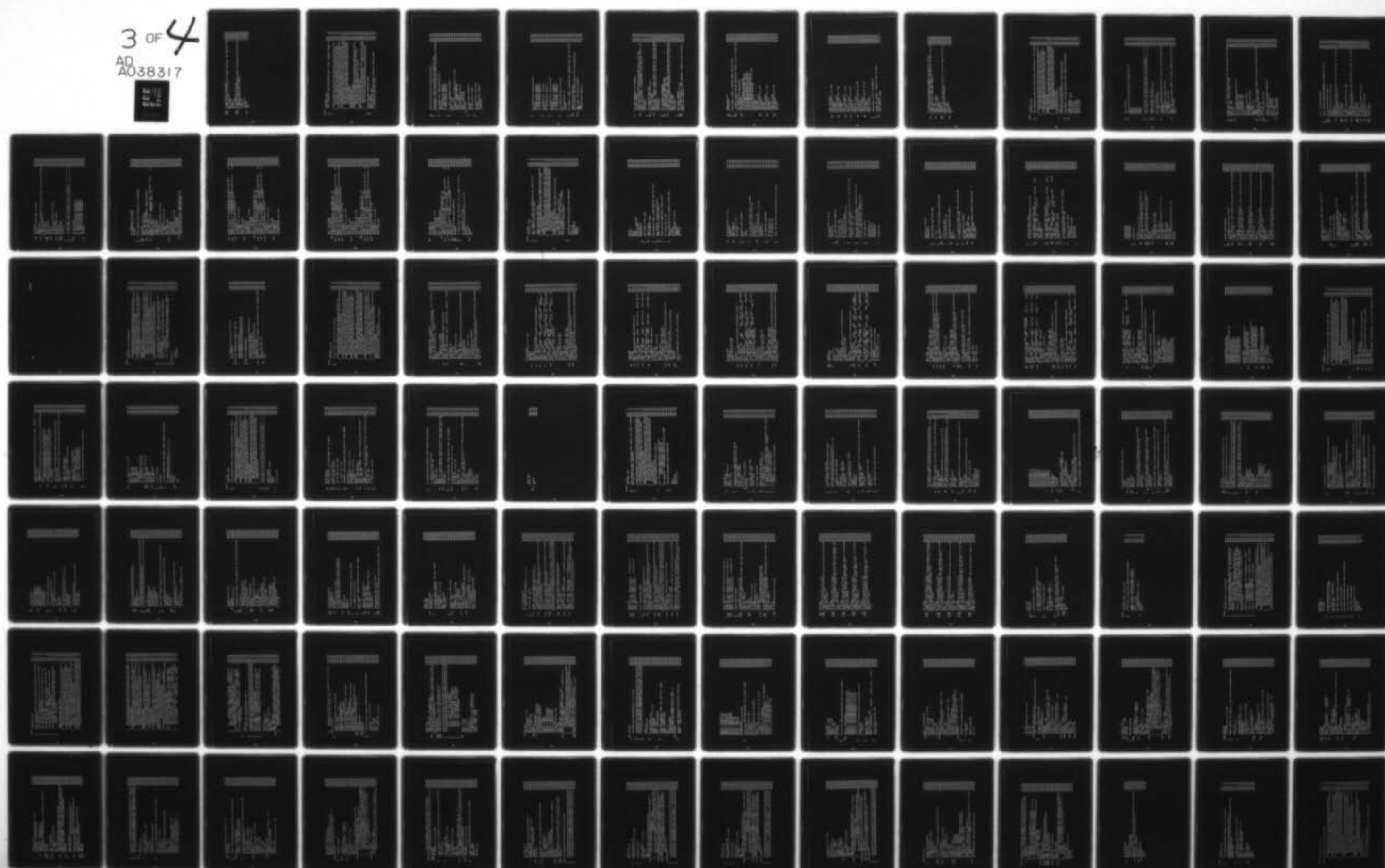
```
310    JJ=J*2                                                     FTAB1790
       KK=K-1                                                     FTAB1800
320    N=6                                                        FTAB1810
       IF (JJ.LT.6) N=JJ                                          FTAB1820
       DO 330 M=1,N                                               FTAB1830
330    DP(M)=DATA(KK+M)                                           FTAB1840
C                                                                 FTAB1850
C      CONVERT OUTPUT DATA UNITS IF NECESSARY.                    FTAB1860
       DO 340 M=2,N,2                                             FTAB1870
  340  DP(M) = DP(M)*XEGSIF                                       FTAB1880
C      IF NO OUTPUT UNITS REQUIRED, SKIP DATA OUTPUT SECTION      FTAB1890
       IF (IPRINT.EQ.0) GO TO 370                                 FTAB1900
C                                                                 FTAB1910
C      OUTPUT DATA                                                FTAB1920
C                                                                 FTAB1930
360    FORMAT (6(5X,1PE12.5))                                     FTAB1940
       WRITE (NPRT,360) (DP(M),M=1,N)                             FTAB1950
       LINE=LINE+1                                                FTAB1960
370    IF (JJ.LE.6) GO TO 410                                     FTAB1970
       IF (IPRINT.EQ.0) GO TO 400                                 FTAB1980
       IF (LINE.LT.NL) GO TO 400                                  FTAB1990
       CALL PAGE                                                  FTAB2000
       WRITE (NPRT,230)                                           FTAB2010
       WRITE (NPRT,260) I,J                                       FTAB2020
       IF(IIUNIT.LE.1) WRITE (NPRT,290)                           FTAB2030
       IF(IIUNIT.GE.2) WRITE (NPRT,280)                           FTAB2040
       LINE = LINE + 5                                            FTAB2050
400    KK=KK+6                                                    FTAB2060
       JJ=JJ-6                                                    FTAB2070
       GO TO 320                                                  FTAB2080
```

```
C
C     CHECK FOR ERRORS AND OUTPUT ERROR MESSAGES
410   L=KDATA(LTABI+2*I-1)
      LL=L+2*KDATA(LTABI+2*I)-2
      DO 420 M=L,LL,2
C
C     CHECK FOR NEGATIVE TIME INPUT
      IF (DATA(M).LT.0.E0) GO TO 430
420   CONTINUE
      GO TO 490
430   IERR=IERR+1
440   FORMAT (1H ,42H*** VALUE OF TIME INPUT FOR FUNCTION TABLE,I5,26H
     1 IS NEGATIVE (FTAB). ***)
      PRINT 440, I
      ITAG=1
450   LINE=LINE+1
      IF (LINE.LT.NL) GO TO 480
      CALL PAGE
      IF (IPRINT.EQ.0) GO TO 490
      WRITE (NPRT,230)
      WRITE (NPRT,260) I,J
      IF (IIUNIT.EQ.0.OR.IIUNIT.EQ.1) GO TO 460
      WRITE (NPRT,280)
      GO TO 470
460   WRITE (NPRT,290)
470   LINE=LINE+5
480   GO TO (490,530,570,590,610), ITAG
490   L=KDATA(LTABI+2*I-1)+2
C
C     CHECK FOR NEGATIVE TIME STEP IN TABLE
```

```
      DO 500 M=L,LL,2                                                   FTAB2390
      IF (DATA(M).LT.DATA(M-2)) GO TO 510                               FTAB2400
500   CONTINUE                                                          FTAB2410
      GO TO 530                                                         FTAB2420
510   IERR=IERR+1                                                       FTAB2430
520   FORMAT (1H ,18H*** FUNCTION TABLE,I5,43H  CONTAINS A NEGATIVE TIMEFTAB2440
     1 STEP (FTAB). ***)                                                FTAB2450
      PRINT 520, I                                                      FTAB2460
      ITAG=2                                                            FTAB2470
      GO TO 450                                                         FTAB2480
C                                                                       FTAB2490
C     RETURN TO PRINT NEXT FUNCTION TABLE                               FTAB2500
530   IF (I.NE.NTAB) GO TO 250                                          FTAB2510
C                                                                       FTAB2520
C     CHECK FOR MULTIPLE OUTPUT                                         FTAB2530
      IF (IIUNIT.EQ.0.OR.IIUNIT.EQ.2) GO TO 550                         FTAB2540
      IF (IERR.GT.IIERR) GO TO 550                                      FTAB2550
      IF (IPRINT.EQ.0) GO TO 550                                        FTAB2560
      XEGSIF=1.E0                                                       FTAB2570
      IF(IIUNIT.EQ.1) IIUNIT = 2                                        FTAB2580
      IF(IIUNIT.EQ.3) IIUNIT = 0                                        FTAB2590
      GO TO 210                                                         FTAB2600
C                                                                       FTAB2610
C     CHECK FOR INPUT TERMINATION DUE TO ERRORS                         FTAB2620
550   IF (JERR.EQ.0) GO TO 570                                          FTAB2630
560   FORMAT (1H ,86H*** FUNCTION TABLE INPUT TERMINATED , FUNCTION TABLFTAB2640
     1ENUMBER OUT OF SEQUENCE (FTAB). ***)                              FTAB2650
      PRINT 560                                                         FTAB2660
      ITAG=3                                                            FTAB2670
      GO TO 450                                                         FTAB2680
```

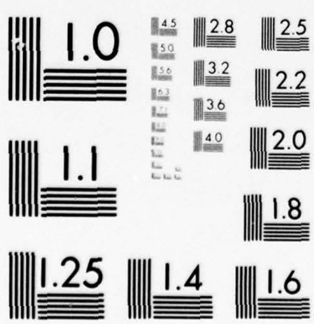
AD-A038 317 VIRGINIA POLYTECHNIC INST AND STATE UNIV BLACKSBURG --ETC F/G 13/13
RELIABILITY STUDY OF SINGER. VOLUME II. USER'S MANUAL.(U)
JAN 77 A E SOMERS, S M HOLZER, J C BRADSHAW F29601-75-C-0050
UNCLASSIFIED AFWL-TR-76-192-VOL-2 NL

3 OF 4
AD
A038317



3 OF 4

38317



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
570   IF (KERR.EQ.0) GO TO 590                                          FTAB2690
580   FORMAT (1H ,68H***INSUFFICIENT STORAGE AVAILABLE FOR FUNCTION TABLFTAB2700
     1ES (FTAB).***      )                                              FTAB2710
      PRINT 580                                                         FTAB2720
      ITAG=4                                                            FTAB2730
      GO TO 450                                                         FTAB2740
590   IF (LERR.EQ.0) GO TO 610                                          FTAB2750
600   FORMAT (1H ,84H***INSUFFICIENT STORAGE AVAILABLE FOR FUNCTION TABLFTAB2760
     1E IDENTIFICATION DATA (FTAB).*** )                                FTAB2770
      PRINT 600                                                         FTAB2780
      ITAG=5                                                            FTAB2790
      GO TO 450                                                         FTAB2800
610   RETURN                                                            FTAB2810
      END                                                               FTAB2820
```

```
CGIDE    0  10
      SUBROUTINE GIDE                                                   GIDE   0
C     THIS SUBROUTINE READS AND CHECKS BLOCK 1 DATA. (CONTROL (GUIDE) DAGIDE  10
C                                                                       GIDE  20
C                                                                       GIDE  30
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), GIDE  40
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),      GIDE  50
     1    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)     GIDE  60
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), GIDE  70
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      GIDE  80
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,GIDE  90
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,GIDE 100
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, GIDE 110
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  GIDE 120
     4    NTIMES,NVEL,IINITD                                            GIDE 130
      COMMON/PRNTBK/MPRINT                                              GIDE 140
      COMMON/SCALE/EGSIF,EGSIL                                          GIDE 150
      COMMON/SEEKBK/DEFOR(90),STPSIZ(90),GRAD(90),GRADI(90),DELTAG(90), GIDE 160
     1    DIRECT(90),DIAG(90),STEP(4),DSTEP(4),FVAL(4),VALUES(7),       GIDE 170
     2    DISACC,SSIZE,FUNACC,FUNMIN,CRITL,CRITU,NLIN                   GIDE 180
      INTEGER HEAD,DHEAD,BBBB,ZZZZ                                      GIDE 190
      DATA SR/1HR/,ES,SE,SS,FI,CF,SM/2HES,2HSE,2HSS,2HFI,1HC,1HM/,DM,EM,GIDE 200
     1      BB,TS/1HD,1HE,1HB,1HS/,BBBB,ZZZZ/4H    ,4H0000/,LAST1,LAST2/ GIDE 210
     2      4HEND ,4HDATA/,AC,BC,CCC,DC,EC/2HCA,2HCB,2HCC,2HCD,2HCE/    GIDE 220
C                                                                       GIDE 230
C     READ TITLE CARD (REQUIRED CARD).                                  GIDE 240
10    FORMAT (20A4)                                                     GIDE 250
      READ (NCRD,10) HEAD                                               GIDE 260
      IF(HEAD(1).EQ.LAST1.AND.HEAD(2).EQ.LAST2) TPROB=-1.0E0            GIDE 270
      IF(TPROB.EQ.-1.0E0) GO TO 400                                     GIDE 280
```

```
20    FORMAT(I5,2(A2,3X),A1,4X,2A1,3X,A1,4X,I5,A1)                         GIDE 290
C                                                                          GIDE 300
C     READ BLOCK 1, CARD 2 OF CONTROL DATA (REQUIRED CARD).                GIDE 310
      READ(NCRD,20) ISTART,UNITS,SLIN,ISTOP,PRINT,PAGEM,STRES,ITAPE,PLOTGIDE 320
C                                                                          GIDE 330
C     SET IUNITS FLAG. 0=EE,1=ES,2=SS,3=SE                                 GIDE 340
      IUNITS = 0                                                           GIDE 350
      IF (UNITS.EQ.ES) IUNITS=1                                            GIDE 360
      IF (UNITS.EQ.SS) IUNITS=2                                            GIDE 370
      IF (UNITS.EQ.SE) IUNITS=3                                            GIDE 380
C                                                                          GIDE 390
C     SET UNITS CONVERSION FACTORS.                                        GIDE 400
      EGSIF = 1.E0                                                         GIDE 410
      EGSIL = 1.E0                                                         GIDE 420
      IF(IUNITS.EQ.0.OR.IUNITS.EQ.2) GO TO 40                              GIDE 430
      EGSIF = 4.4482216152605E0                                            GIDE 440
      EGSIL = .0254E0                                                      GIDE 450
      IF(IUNITS.EQ.1) GO TO 40                                             GIDE 460
      EGSIF = 1.E0/EGSIF                                                   GIDE 470
      EGSIL = 1.E0/EGSIL                                                   GIDE 480
   40 CONTINUE                                                             GIDE 490
C                                                                          GIDE 500
C     SET ILIN FLAG. 0=II,1=FI.                                            GIDE 510
      ILIN=0                                                               GIDE 520
      IF (SLIN.EQ.FI) ILIN=1                                               GIDE 530
C                                                                          GIDE 540
C     SET ISTOP FLAG. 0=C,1=STOP                                           GIDE 550
      IFLAG=1                                                              GIDE 560
      IF (ISTOP.EQ.CF) IFLAG=0                                             GIDE 570
      ISTOP=IFLAG                                                          GIDE 580
```

```
C                                                                         GIDE 590
C     SET IPRINT FLAG.  0=M,1=S,2=D,3=E                                   GIDE 600
      IPRINT=1                                                            GIDE 610
      IF (PRINT.EQ.SM) IPRINT=0                                           GIDE 620
      IF (PRINT.EQ.DM) IPRINT=2                                           GIDE 630
      IF (PRINT.EQ.EM) IPRINT=3                                           GIDE 640
C                                                                         GIDE 650
C     SET MPRINT FLAG.   0=M,1=S.                                         GIDE 660
      MPRINT = 1                                                          GIDE 670
      IF (PAGEM.EQ.SM) MPRINT = 0                                         GIDE 680
C                                                                         GIDE 690
C     SET ISTRES FLAG. 0=N,1=S,2=R,3=B                                    GIDE 700
      ISTRES=0                                                            GIDE 710
      IF (STRES.EQ.TS) ISTRES=1                                           GIDE 720
      IF (STRES.EQ.SR) ISTRES=2                                           GIDE 730
      IF (STRES.EQ.BB) ISTRES=3                                           GIDE 740
C                                                                         GIDE 750
C     SET IPLOT=0, NO PLOT,, =1, PLOT.                                    GIDE 760
      N=0                                                                 GIDE 770
      IF (PLOT.EQ.SR) N=1                                                 GIDE 780
      IPLOT=N                                                             GIDE 790
C                                                                         GIDE 800
C     READ BLOCK 1, CARD 3 SOLUTION DATA (REQUIRED CARD) AND TEST DATA.   GIDE 810
C                                                                         GIDE 820
220   FORMAT (A1,7X,I2,5E10.0)                                            GIDE 830
      READ (NCRD,220) ANAL,IINITD,TBEGIN,THALT,TINK,SERR,TPROB            GIDE 840
C                                                                         GIDE 850
C SET IANAL FLAG. (0=STATIC,1=DYNAMIC)                                    GIDE 860
      IANAL = 0                                                           GIDE 870
      IF (ANAL.EQ.DM) IANAL = 1                                           GIDE 880
```

```
C
C     TEST FOR IMPOSSIBLE TIME INTERVAL.
      TTT=THALT-TBEGIN
      IF(TTT.GT.0.E0.OR.(TTT.EQ.0.E0.AND.IANAL.EQ.0)) GO TO 250
240   FORMAT (55H ***BLOCK 1 INPUT INDICATES AN INTEGRATION INTERVAL OF
     1,0PE12.4,44H SECONDS, AN UNACCEPTABLE VALUE (GIDE).***   )
      PRINT 240, TTT
      IERR=IERR+1
C
C     TEST PROBLEM TIME LIMIT.
250   IF (TPROB.GT.0.E0) GO TO 260
      PRINT 230, TPROB
230   FORMAT (38H  **BLOCK 1 INPUT ADMITS A MAXIMUM OF ,0PE10.0,70HCOMPU
     1TER MINUTES.  DEFAULT PROBLEM TIME IS SET TO 120 MINUTES(GIDE).**)
      TPROB=120.E0
      IREC=IREC+1
C
C     TEST ACCURACY REQUIREMENT.
260   IF (SERR.EQ.0.E0) SERR=SQRT(EPS)
      IF (SERR.GE.SQRT(EPS)) GO TO 280
270   FORMAT (57H ***BLOCK 1 INPUT SEEKS A RELATIVE ENERGY ERROR LESS TH
     1AN ,1PE10.3,50H COMPUTER PRECISION LIMITS PREVENT THIS (GIDE).***)
      RERR = SQRT(EPS)
      PRINT 270, RERR
      IERR=IERR+1
C
C     TEST PRINT INTERVAL.
280   IF (TINK.GE.0.E0) GO TO 300
290   FORMAT (54H ***BLOCK 1 INPUT REQUIRES A NEGATIVE PRINT INTERVAL, ,
     10PE10.0,12H (GIDE).*** )
```

```
      PRINT 290, TINK                                                   GIDE1190
300   DISACC=SERR                                                       GIDE1200
C                                                                       GIDE1210
C     READ BLOCK 1, CARD 4 (REQUIRED CARD) AND TEST DATA.               GIDE1220
C                                                                       GIDE1230
320    FORMAT(1H ,23H** FAILURE COEFFICIENT ,A2,25H IS OUT OF RANGE, SETGIDE1240
     1 TO ,E9.2,3H **)                                                  GIDE1250
      READ (NCRD,221) CA,CB,CC,CD,CE                                    GIDE1260
221   FORMAT(5E10.0)                                                    GIDE1270
C     SET DEFAULT VALUES IF NECESSARY                                   GIDE1280
      IF(CA.EQ.BBBB.OR.CA.EQ.0.E0) CA=1.E0                              GIDE1290
      IF(CB.EQ.BBBB.OR.CB.EQ.0.E0) CB=2.E0                              GIDE1300
      IF(CC.EQ.BBBB.OR.CC.EQ.0.E0) CC=3.5E0                             GIDE1310
      IF(CD.EQ.BBBB.OR.CD.EQ.0.E0)CD=0.E0                               GIDE1320
      IF(CE.EQ.BBBB.OR.CE.EQ.0.E0) CE=4.E0/3.E0                         GIDE1330
      IF(CA.LE.1.23E0) GO TO 322                                        GIDE1340
      IREC=IREC+1                                                       GIDE1350
      CA=1.23E0                                                         GIDE1360
      PRINT 320,AC,CA                                                   GIDE1370
322   IF(CA.GE.1.E0) GO TO 324                                          GIDE1380
      IREC=IREC+1                                                       GIDE1390
      CA=1.E0                                                           GIDE1400
      PRINT 320,AC,CA                                                   GIDE1410
324   IF(CB.LE.4.E0) GO TO 326                                          GIDE1420
      IREC=IREC+1                                                       GIDE1430
      CB=4.E0                                                           GIDE1440
      PRINT 320,BC,CB                                                   GIDE1450
326   IF(CB.GE.1.E0) GO TO 328                                          GIDE1460
      IREC=IREC+1                                                       GIDE1470
      CB=1.E0                                                           GIDE1480
```

```
      PRINT 320,BC,CB                                                   GIDE1490
328   IF(CC.LE.5.83E0) GO TO 330                                        GIDE1500
      IREC=IREC+1                                                       GIDE1510
      CC=5.83E0                                                         GIDE1520
      PRINT 320,CCC,CC                                                  GIDE1530
330   IF(CC.GE.3.5E0) GO TO 332                                         GIDE1540
      IREC=IREC+1                                                       GIDE1550
      CC=3.5E0                                                          GIDE1560
      PRINT 320,CCC,CC                                                  GIDE1570
332   IF(CD.LE.1.E0) GO TO 334                                          GIDE1580
      CD=1.E0                                                           GIDE1590
      PRINT 320,DC,CD                                                   GIDE1600
334   IF(CD.GE.0.E0) GO TO 336                                          GIDE1610
      IREC=IREC+1                                                       GIDE1620
      CD=0.E0                                                           GIDE1630
      PRINT 320,DC,CD                                                   GIDE1640
336   IF(CE.LE.4.E0/3.E0) GO TO 338                                     GIDE1650
      IREC=IREC+1                                                       GIDE1660
      CE=4.E0/3.E0                                                      GIDE1670
      PRINT 320,EC,CE                                                   GIDE1680
338   IF(CE.GE.4.E0/7.E0) GO TO 340                                     GIDE1690
      IREC=IREC+1                                                       GIDE1700
      CE=4.E0/7.E0                                                      GIDE1710
      PRINT 320,EC,CE                                                   GIDE1720
C                                                                       GIDE1730
C     READ BLOCK 1, CARD 5 (REQUIRED).                                  GIDE1740
C                                                                       GIDE1750
C     SEARCH FOR A BLANK ^ ASSUME IT ENDS THE BLOCK.                    GIDE1760
340   DO 380 I=1,100                                                    GIDE1770
      READ (NCRD,10) DHEAD                                              GIDE1780
```

```
      IF ((DHEAD(1).EQ.BBBB).OR.(DHEAD(1).EQ.ZZZZ)) GO TO 400           GIDE1790
350   FORMAT (95H ***AT LEAST ONE CARD IS OUT-OF-SORT. THE NEXT CARD WASGIDE1800
     1 EXPECTED TO BE A ZERO CARD (GIDE).***  )                         GIDE1810
      PRINT 350                                                         GIDE1820
370   FORMAT (1H ,20A4)                                                 GIDE1830
      PRINT 370, DHEAD                                                  GIDE1840
      IERR=IERR+1                                                       GIDE1850
380   CONTINUE                                                          GIDE1860
390   FORMAT (84H ***A ZEROS CARD FOR BLOCK 1 WAS NOT FOUND AFTER 100 CAGIDE1870
     1RDS HAD BEEN READ (GIDE)***  )                                    GIDE1880
      PRINT 390                                                         GIDE1890
      IERR=IERR+1                                                       GIDE1900
400   RETURN                                                            GIDE1910
      END                                                               GIDE1920
```

```
CINIT    0  10
      SUBROUTINE INIT
C
C     THIS SUBROUTINE READS AND CHECKS INITIAL CONDITIONS AT THE JOINTS.
C
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50),
     1     ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),
     1     XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20),
     1     PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB
      INTEGER HEAD,DHEAD
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,
     1     IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,
     2     NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,
     3     NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,
     4     NTIMES,NVEL,IINITD
      COMMON/SCALE/EGSIF,EGSIL
C
      DIMENSION A(3),ND(6),MD(3)
      DATA ND/1HV,1HA,1HJ,1HF,1HD,1H /,MD/2HDX,2HDY,2HRZ/
C
C     READ USER*S TITLE AND INITIALIZE JOINT ARRAYS AND COUNTERS.
C
10    FORMAT (20A4)
      READ (NCRD,10) DHEAD
      DO 20 J=1,NJ
      DO 20 K=1,3
      DIS(K,J)=TINY
      VEL(K,J)=TINY
      ACC(K,J)=TINY
```

```
20    BET(K,J)=TINY                                                        INIT 290
C                                                                          INIT 300
C     FOR(K,J) AND IFOR INITIALIZED IN SUBROUTINE BODY.                    INIT 310
      NINC=0                                                               INIT 320
      JERR=0                                                               INIT 330
      NDIS=0                                                               INIT 340
      NVEL=0                                                               INIT 350
      NACC=0                                                               INIT 360
      NJER=0                                                               INIT 370
      NJOR=0                                                               INIT 380
C                                                                          INIT 390
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *INIT 400
C     READ JOINT NO. AND INITIAL CONDITION TYPE AND COMPONENT.             INIT 410
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *INIT 420
C                                                                          INIT 430
30    FORMAT (I5,A1,4X,3E10.0)                                             INIT 440
40    READ (NCRD,30) J,L,(A(I),I=1,3)                                      INIT 450
C                                                                          INIT 460
C     TEST FOR LAST CARD.                                                  INIT 470
      IF (J.EQ.0) GO TO 340                                                INIT 480
C                                                                          INIT 490
C     INCREMENT COUNTER AND CHECK RANGE OF JOINT NO. AND TYPE INDICATOR.INIT 500
      NINC=NINC+1                                                          INIT 510
      IF (J.GE.1.AND.J.LE.NJ) GO TO 80                                     INIT 520
      IERR=IERR+1                                                          INIT 530
      IF (JERR.EQ.1) GO TO 70                                              INIT 540
      CALL PAGE                                                            INIT 550
50    FORMAT (1H ,10X,44HINPUT ERRORS IN INITIAL CONDITION DATA BLOCK//)INIT 560
      WRITE (NPRT,50)                                                      INIT 570
      JERR=1                                                               INIT 580
```

```
60     FORMAT (14H *** JOINT NO.,I5,50H LESS THAN ONE OR GREATER THAN LARINIT 590
      1GEST JOINT NO. =,I4,12H (INIT). ***/)                              INIT 600
70     PRINT 60, J,NJ                                                     INIT 610
       GO TO 40                                                           INIT 620
80     LA = 0                                                             INIT 630
       IF (L.EQ.ND(5).OR.L.EQ.ND(6)) LA=1                                 INIT 640
       IF (L.EQ.ND(1)) LA=2                                               INIT 650
       IF (L.EQ.ND(2)) LA=3                                               INIT 660
       IF (L.EQ.ND(3)) LA=4                                               INIT 670
       IF (L.EQ.ND(4)) LA=5                                               INIT 680
       L=LA                                                               INIT 690
       IF (L.NE.0) GO TO 110                                              INIT 700
       IERR=IERR+1                                                        INIT 710
       IF (JERR.EQ.1) GO TO 100                                           INIT 720
       CALL PAGE                                                          INIT 730
       WRITE (NPRT,50)                                                    INIT 740
       JERR=1                                                             INIT 750
90     FORMAT (1H ,26H*** INITIAL CONDITION TYPE,I5,9H AT JOINT,I5,27H NOINIT 760
      1T ACCEPTABLE (INIT). ***)                                          INIT 770
100    PRINT 90, L,J                                                      INIT 780
       GO TO 40                                                           INIT 790
110    IF (NJ.GT.NJD) GO TO 40                                            INIT 800
C                                                                         INIT 810
C      CHANGE UNITS OF INITIAL CONDITIONS.                                INIT 820
       IF (L.EQ.5) GO TO 120                                              INIT 830
       A(1) = A(1)*EGSIL                                                  INIT 840
       A(2) = A(2)*EGSIL                                                  INIT 850
       GO TO 130                                                          INIT 860
   120 A(1) = A(1)*EGSIF                                                  INIT 870
       A(2) = A(2)*EGSIF                                                  INIT 880
```

```
      A(3) = A(3)*(EGSIF*EGSIL)                                         INIT 890
C                                                                       INIT 900
C     STORE INITIAL CONDITION COMPONENT ACCORDING TO TYPE.              INIT 910
C                                                                       INIT 920
130   GO TO (140,180,220,260,300), L                                    INIT 930
140   IFOR=1                                                            INIT 940
      IF (DIS(1,J).EQ.TINY) GO TO 160                                   INIT 950
      JERR=1                                                            INIT 960
      IERR=IERR+1                                                       INIT 970
150   FORMAT (1H ,12H***JOINT NO.,I5,60H HAS MORE THAN ONE SET OF INITIAINIT 980
     1L DISPLACEMENTS (INIT).***   )                                    INIT 990
      PRINT 150, J                                                      INIT1000
160   DO 170 I=1,3                                                      INIT1010
170   DIS(I,J)=A(I)                                                     INIT1020
      GO TO 40                                                          INIT1030
180   IF (VEL(1,J).EQ.TINY) GO TO 200                                   INIT1040
      JERR=1                                                            INIT1050
      IERR=IERR+1                                                       INIT1060
190   FORMAT (1H ,12H***JOINT NO.,I5,60H HAS MORE THAN ONE SET OF INITIAINIT1070
     1L VELOCITIES (INIT).***      )                                    INIT1080
      PRINT 190, J                                                      INIT1090
200   DO 210 I=1,3                                                      INIT1100
210   VEL(I,J)=A(I)                                                     INIT1110
      GO TO 40                                                          INIT1120
220   IF (ACC(1,J).EQ.TINY) GO TO 240                                   INIT1130
      DO 230 I=1,3                                                      INIT1140
230   ACC(I,J)=A(I)+ACC(I,J)                                            INIT1150
      GO TO 40                                                          INIT1160
240   DO 250 I=1,3                                                      INIT1170
250   ACC(I,J)=A(I)                                                     INIT1180
```

```
      GO TO 40                                                          INIT1190
260   IF (BET(1,J).EQ.TINY) GO TO 280                                   INIT1200
      JERR=1                                                            INIT1210
      IERR=IERR+1                                                       INIT1220
270   FORMAT (1H ,12H***JOINT NO.,I5,60H HAS MORE THAN ONE SET OF INITIAINIT1230
     1L JERKS (INIT).***             )                                  INIT1240
      PRINT 270, J                                                      INIT1250
280   DO 290 I=1,3                                                      INIT1260
290   BET(I,J)=A(I)                                                     INIT1270
      GO TO 40                                                          INIT1280
300   IFOR=1                                                            INIT1290
      IF (FOR(1,J).EQ.TINY) GO TO 320                                   INIT1300
      DO 310 I=1,3                                                      INIT1310
310   FOR(I,J)=FOR(I,J)+A(I)                                            INIT1320
      GO TO 40                                                          INIT1330
320   DO 330 K=1,3                                                      INIT1340
330   FOR(K,J)=A(K)                                                     INIT1350
      GO TO 40                                                          INIT1360
C                                                                       INIT1370
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *INIT1380
C     IF LAST CARD, PRINT OUT INPUT DATA, IF REQUIRED.                  INIT1390
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *INIT1400
C                                                                       INIT1410
340   DO 350 I=1,NJ                                                     INIT1420
      IF (DIS(1,I).NE.TINY)  NDIS=NDIS+1                                INIT1430
      IF (VEL(1,I).NE.TINY)  NVEL=NVEL+1                                INIT1440
      IF (ACC(1,I).NE.TINY)  NACC=NACC+1                                INIT1450
      IF (BET(1,I).NE.TINY)  NJER=NJER+1                                INIT1460
      IF (FOR(1,I).NE.TINY)  NJOR=NJOR+1                                INIT1470
350   CONTINUE                                                          INIT1480
```

```
      IF (JERR.NE.0) GO TO 670
      IF (NINC.EQ.0) GO TO 670
      NPAGE=1
C
C     PRINT HEADING WITH EITHER EQ OR SI DIMENSIONS
C
360   CALL PAGE
370   FORMAT (1H ,20A4,//)
      WRITE (NPRT,370) DHEAD
380   FORMAT (1H ,33X,24HINITIAL  CONDITION DATA /)
      WRITE (NPRT,380)
390   FORMAT (1H ,92H JOINT  CONDITION TYPE  X COMPONENT  UNITS      Y C
     1OMPONENT  UNITS       Z COMPONENT  UNITS  /)
      WRITE (NPRT,390)
      DO 660 J=1,NJ
      IN=NPAGE*IUNITS
      ILINE=LINE
      IF (DIS(1,J).EQ.TINY) GO TO 440
      LINE=LINE+1
      IF (LINE.LT.NL) GO TO 400
      CALL PAGE
      WRITE (NPRT,390)
400   A(1)=DIS(1,J)
      A(2)=DIS(2,J)
      A(3)=DIS(3,J)
      IF(NPAGE.EQ.2) GO TO 430
      A(1) = A(1)/EGSIL
      A(2) = A(2)/EGSIL
  430 IF(IN.GE.2.AND.IN.LE.4) WRITE(NPRT,410) J,A
      IF(IN.LT.2.OR.IN.GT.4) WRITE(NPRT,420) J, A
```

```
410    FORMAT (1H ,I5,2X,14H DISPLACEMENT ,2X,0PE12.5,2X,10H M          ,1PINIT1790
      1E12.5,1X,9H M        ,2X,0PE12.5,2X,9H RAD       )                  INIT1800
420    FORMAT (1H ,I5,2X,14H DISPLACEMENT ,2X,0PE12.5,2X,10H IN.        ,1PINIT1810
      1E12.5,1X,9H IN.      ,2X,0PE12.5,2X,9H RAD       )                  INIT1820
440    IF (VEL(1,J).EQ.TINY) GO TO 490                                     INIT1830
       LINE=LINE+1                                                         INIT1840
       IF (LINE.LT.NL) GO TO 450                                           INIT1850
       CALL PAGE                                                           INIT1860
       WRITE (NPRT,390)                                                    INIT1870
450    A(1)=VEL(1,J)                                                       INIT1880
       A(2)=VEL(2,J)                                                       INIT1890
       A(3)=VEL(3,J)                                                       INIT1900
       IF(NPAGE.EQ.2) GO TO 480                                            INIT1910
       A(1) = A(1)/EGSIL                                                   INIT1920
       A(2) = A(2)/EGSIL                                                   INIT1930
  480 IF(IN.GE.2.AND.IN.LE.4) WRITE(NPRT,460) J,A                          INIT1940
       IF(IN.LT.2.OR.IN.GT.4) WRITE(NPRT,470) J,A                          INIT1950
460    FORMAT (1H ,I5,2X,14H VELOCITY     ,2X,0PE12.5,2X,10H M/S        ,1PINIT1960
      1E12.5,1X,9H M        ,2X,0PE12.5,2X,9H RAD/S   )                    INIT1970
470    FORMAT (1H ,I5,2X,14H VELOCITY     ,2X,0PE12.5,2X,10H IN./S      ,1PINIT1980
      1E12.5,1X,9H IN./S    ,2X,0PE12.5,2X,9H RAD/S   )                    INIT1990
490    IF (ACC(1,J).EQ.TINY) GO TO 540                                     INIT2000
       LINE=LINE+1                                                         INIT2010
       IF (LINE.LT.NL) GO TO 500                                           INIT2020
       CALL PAGE                                                           INIT2030
       WRITE (NPRT,390)                                                    INIT2040
500    A(1)=ACC(1,J)                                                       INIT2050
       A(2)=ACC(2,J)                                                       INIT2060
       A(3)=ACC(3,J)                                                       INIT2070
       IF(NPAGE.EQ.2) GO TO 530                                            INIT2080
```

```
      A(1) = A(1)/EGSIL                                                 INIT2090
      A(2) = A(2)/EGSIL                                                 INIT2100
  530 IF(IN.GE.2.AND.IN.LE.4) WRITE(NPRT,510) J,A                       INIT2110
      IF(IN.LT.2.OR.IN.GT.4) WRITE(NPRT,520) J,A                        INIT2120
510   FORMAT (1H ,I5,2X,14H ACCELERATION ,2X,0PE12.5,2X,10H M/S**2    ,1PINIT2130
     1E12.5,1X,9H M/S**2  ,2X,0PE12.5,2X,9H RAD /S**2)                  INIT2140
520   FORMAT (1H ,I5,2X,14H ACCELERATION ,2X,0PE12.5,2X,10H IN./S**2 ,1PINIT2150
     1E12.5,1X,9H IN./S**2,2X,0PE12.5,2X,9H RAD/S**2)                   INIT2160
540   IF (BET(1,J).EQ.TINY) GO TO 590                                   INIT2170
      LINE=LINE+1                                                       INIT2180
      IF (LINE.LT.NL) GO TO 550                                         INIT2190
      CALL PAGE                                                         INIT2200
      WRITE (NPRT,390)                                                  INIT2210
550   A(1)=BET(1,J)                                                     INIT2220
      A(2)=BET(2,J)                                                     INIT2230
      A(3)=BET(3,J)                                                     INIT2240
      IF(NPAGE.EQ.2) GO TO 580                                          INIT2250
      A(1) = A(1)/EGSIL                                                 INIT2260
      A(2) = A(2)/EGSIL                                                 INIT2270
  580 IF(IN.GE.2.AND.IN.LE.4) WRITE(NPRT,560) J,A                       INIT2280
      IF(IN.LT.2.OR.IN.GT.4) WRITE(NPRT,570) J, A                       INIT2290
560   FORMAT (1H ,I5,2X,14H JERK         ,2X,0PE12.5,2X,10H M/S**3    ,1PINIT2300
     1E12.5,1X,9H M/S**3  ,0PE12.5,2X,9H RAD/S**3)                      INIT2310
570   FORMAT (1H ,I5,2X,14H JERK         ,2X,0PE12.5,2X,10H IN./S**3 ,1PINIT2320
     1E12.5,1X,9H IN./S**3,0PE12.5,2X,9H RAD/S**3)                      INIT2330
590   IF (FOR(1,J).EQ.TINY) GO TO 650                                   INIT2340
      LINE=LINE+1                                                       INIT2350
      IF (LINE.LT.NL) GO TO 600                                         INIT2360
      CALL PAGE                                                         INIT2370
      WRITE (NPRT,390)                                                  INIT2380
```

```
600    A(1)=FOR(1,J)                                                        INIT2390
       A(2)=FOR(2,J)                                                        INIT2400
       A(3)=FOR(3,J)                                                        INIT2410
       IF(NPAGE.EQ.2) GO TO 630                                             INIT2420
       A(1) = A(1)/EGSIF                                                    INIT2430
       A(2) = A(2)/EGSIF                                                    INIT2440
       A(3) = A(3)/(EGSIF*EGSIL)                                            INIT2450
  630  IF(IN.GE.2.AND.IN.LE.4) WRITE(NPRT,610) J,A                          INIT2460
       IF(IN.LT.2.OR.IN.GT.4) WRITE(NPRT,620) J,A                           INIT2470
610    FORMAT (1H ,I5,2X,14H LOAD         ,2X,0PE12.5,2X,10H N         ,1PINIT2480
      1E12.5,1X,9H N        ,2X,0PE12.5,2X,9H N*M      )                INIT2490
620    FORMAT (1H ,I5,2X,14H LOAD         ,2X,0PE12.5,2X,10H LB        ,1PINIT2500
      1E12.5,1X,9H LB       ,2X,0PE12.5,2X,9H LB*IN.   )                INIT2510
640    FORMAT (1H )                                                         INIT2520
650    IF (LINE.GT.ILINE) WRITE (NPRT,640)                                  INIT2530
660    CONTINUE                                                             INIT2540
C                                                                           INIT2550
C      IF INPUT-OUTPUT UNITS ARE MIXED, PRINT SECOND SET OF DATA.           INIT2560
C                                                                           INIT2570
       IF (NPAGE.EQ.2) GO TO 670                                            INIT2580
       IF (IUNITS.EQ.0.OR.IUNITS.EQ.2) GO TO 670                            INIT2590
       NPAGE=2                                                              INIT2600
       GO TO 360                                                            INIT2610
670    RETURN                                                               INIT2620
       END                                                                  INIT2630
```

```
CJFCR    0  10
      SUBROUTINE JFOR                                                   JFOR
C                                                                       JFOR
C     THIS SUBROUTINE  READS AND ERROR CHECKS DATA INPUT TO THE         JFOR
C     JOINT FORCING FUNCTION DATA BLOCK.                                JFOR
C                                                                       JFOR
      COMMON DATA(10000),KDATA(500)                                     JFOR
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), JFOR
     1   PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB       JFOR
      INTEGER HEAD,DHEAD                                                JFOR
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,JFOR
     1   IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM, JFOR
     2   NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,  JFOR
     3   NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,   JFOR
     4   NTIMES,NVEL,IINITO                                             JFOR
      COMMON/SCALE/EGSIF,EGSIL                                          JFOR
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,          JFOR
     1   LTABI,NMAX,NMAXI                                               JFOR
      DIMENSION IERROR(50,5),AD(4),AK(3)                                JFOR
      DATA EX,WY,ZZ,SI,CS/1HX,1HY,1HZ,1HS,1HC/                          JFOR
      DATA AD/4HNORM,4H  X ,4H  Y ,4H  Z /,BBB/4H    /                  JFOR
      DATA AK/4HCOS ,4HSINE,4HCNST/                                     JFOR
C                                                                       JFOR
C                                                                       JFOR
C     INITIALIZE VARIABLES USED BY THIS SUBROUTINE                      JFOR
      I=0                                                               JFOR
      IIERR=IERR                                                        JFOR
      IIUNIT=IUNITS                                                     JFOR
      JERROR=0                                                          JFOR
      KERROR=0                                                          JFOR
```

```
      LP=LFF                                                            JFOR 290
      LPI=LFFI                                                          JFOR 300
      NDL=0                                                             JFOR 310
      NFF=0                                                             JFOR 320
      XEGSIF = 1.E0/EGSIF                                               JFOR 330
      DO 10 M=1,NJD                                                     JFOR 340
      DO 10 N=1,5                                                       JFOR 350
10    IERROR(M,N)=0                                                     JFOR 360
C                                                                       JFOR 370
C     READ BLOCK 8 TITLE CARD.                                          JFOR 380
C                                                                       JFOR 390
30    FORMAT (20A4)                                                     JFOR 400
      READ (NCRD,30) DHEAD                                              JFOR 410
C                                                                       JFOR 420
C     READ BLOCK 8, CARD 2 (NCNE REQUIRED) AND TEST DATA.               JFOR 430
C                                                                       JFOR 440
40    READ (NCRD,50) J,JA,AY,LA,AL,A,B,C,D                              JFOR 450
50    FORMAT (2I5,A1,4X,I4,A1,4E10.0)                                   JFOR 460
C                                                                       JFOR 470
C     CHECK FOR LAST CARD INPUT TO DATA BLOCK                           JFOR 480
C                                                                       JFOR 490
      IF (J.EQ.0) GO TO 180                                             JFOR 500
C                                                                       JFOR 510
C     CHECK AVAILABILITY OF STORAGE TO STORE DATA                       JFOR 520
      LL=LP+4                                                           JFOR 530
      LLI=LPI+5                                                         JFOR 540
      IF (LL.LE.NMAX) GO TO 60                                          JFOR 550
      IERR=IERR+1                                                       JFOR 560
      JERROR=1                                                          JFOR 570
      GO TO 70                                                          JFOR 580
```

```
60    IF (LLI.LE.NMAXI) GO TO 90
      IERR=IERR+1
      KERROR=1
70    READ (NCRD,80) J
80    FORMAT (I5)
      IF (J.NE.0) GO TO 70
      GO TO 180
C
C     INCREMENT JOINT COUNTER
90    I=I+1
      LP=LL
      LPI=LLI
C
C     CHECK THAT JOINT NUMBER INPUT IS DEFINED
      IF (J.GT.0.AND.J.LE.NJ) GO TO 100
      IERR=IERR+1
      IERROR(I,1)=1
100   IF (JA.EQ.0) GO TO 110
      IF (JA.GT.0.AND.JA.LE.NJ) GO TO 110
      IERR=IERR+1
      IERROR(I,1)=1
C
C     SET DIRECTION PARAMETERS.
110   K = 1
      IF (AY.EQ.EX) K=2
      IF (AY.EQ.WY) K=3
      IF (AY.EQ.ZZ) K=4
C
C     CHECK THAT REFERENCE FUNCTION SPECIFIED IS DEFINED
C
```

```
      IF (LA.LE.NTAB) GO TO 120
      IERR=IERR+1
      IERROR(I,3)=1
C
C     SET FUNCTION TYPE PARAMETER.
120   L = -1
      IF (LA.GT.0) L=LA
      IF (AL.EQ.SI) L=-2
      IF (AL.EQ.CS) L=-3
C     SET SCALE FACTOR DEFAULT
      IF(A.EQ.BBB.OR.A.EQ.0.E0)A=1.E0
C
C     CHECK THAT A LOADED JOINT IS GIVEN FOR A DISTRIBUTED LOAD.
      IF(JA.EQ.0) NJOR = NJOR + 1
      IF(JA.NE.0) NDL = NDL + 1
C
C     CHECK THAT TIME PERIOD INPUT IS COMPATIBLE WITH THE
C     REFERENCE FUNCTION SPECIFIED.
      IF(L.GE.-1.AND.D.NE.0.E0) IERROR(I,4) = 1
      IF(L.LT.-1.AND.D.LE.0.E0) IERROR(I,5) = 1
C
C     STORE INPUT DATA
C
      KDATA(LPI-4) = J
      KDATA(LPI-3)=JA
      KDATA(LPI-2)=K
      KDATA(LPI-1)=L
      KDATA(LPI)=2
C
      DATA(LP-3)=A
```

```
      DATA(LP-2) = B*EGSIF                                              JFOR1190
      DATA(LP-1)=C                                                      JFOR1200
      DATA(LP)=D                                                        JFOR1210
C                                                                       JFOR1220
C     RETURN TO READ NEXT INPUT CARD                                    JFOR1230
C                                                                       JFOR1240
      GO TO 40                                                          JFOR1250
C                                                                       JFOR1260
C     SET NUMBER OF JOINT FORCING FUNCTIONS INPUT                       JFOR1270
180   NFF=I                                                             JFOR1280
C                                                                       JFOR1290
C     CHECK FOR NO INPUT                                                JFOR1300
      IF (NFF.EQ.0) GO TO 500                                           JFOR1310
C                                                                       JFOR1320
C     INITIALIZE COUNTERS FOR OUTPUT                                    JFOR1330
190   LL=LFF                                                            JFOR1340
      MM=LFFI                                                           JFOR1350
      I=0                                                               JFOR1360
C                                                                       JFOR1370
C     IF NO OUTPUT REQUIRED, SKIP DATA OUTPUT SECTION                   JFOR1380
      IF (IPRINT.EQ.0) GO TO 280                                        JFOR1390
C                                                                       JFOR1400
C     PRINT DATA BLOCK HEADING                                          JFOR1410
      CALL PAGE                                                         JFOR1420
200   FORMAT (1H ,20A4)                                                 JFOR1430
      WRITE (NPRT,200) DHEAD                                            JFOR1440
      LINE=LINE+1                                                       JFOR1450
220   FORMAT (1H0,25X,30HJOINT FORCING FUNCTION DATA   /)               JFOR1460
      WRITE (NPRT,220)                                                  JFOR1470
      LINE=LINE+2                                                       JFOR1480
```

```
C
C     CHECK FOR OUTPUT UNITS
      IF (IIUNIT.EQ.0.OR.IIUNIT.EQ.1) GO TO 250
C
C     OUTPUT HEADINGS AND METRIC UNITS
230   FORMAT (1H0,90HJOINT  REFERENCE LOAD (N)  FUNCTION    SCALE
     1  LOAD (N)       TIME (S)       TIME (S) )
      WRITE (NPRT,230)
      LINE=LINE+3
      GO TO 270
C
C     OUTPUT HEADINGS AND ENGLISH UNITS
240   FORMAT (1H0,90HJOINT  REFERENCE LOAD (LB) FUNCTION    SCALE
     1  LOAD (LB)      TIME (S)       TIME (S) )
250   WRITE (NPRT,240)
260   FORMAT (1H ,90HLOADED JOINT    DIRECTION NUMBER      FACTOR
     1  ADDITION       ADDITION       PERIOD    ,/)
270   WRITE (NPRT,260)
C     INCREMENT OUTPUT COUNTER
280   I=I+1
C
C     IF NO OUTPUT REQUIRED, SKIP DATA OUTPUT SECTION
      IF (IPRINT.EQ.0) GO TO 380
C
C     PREPARE DATA FOR OUTPUT
      A=DATA(LL+1)
      B = DATA(LL+2)*XEGSIF
      C=DATA(LL+3)
      D=DATA(LL+4)
C
```

```
      J=KDATA(MM+1)                                                     JFOR1790
      JA=KDATA(MM+2)                                                    JFOR1800
      K=KDATA(MM+3)                                                     JFOR1810
      L=KDATA(MM+4)                                                     JFOR1820
C                                                                       JFOR1830
C     OUTPUT DATA                                                       JFOR1840
C                                                                       JFOR1850
      L1=L+4                                                            JFOR1860
      IF (L1.LE.3) GO TO 310                                            JFOR1870
290   FORMAT (1H ,I5,3X,I5,6X,A4,4X,I4,5X,4(0PE14.5))                   JFOR1880
      WRITE (NPRT,290) J,JA,AD(K),L,A,B,C,D                             JFOR1890
      GO TO 320                                                         JFOR1900
300   FORMAT (1H ,I5,3X,I5,6X,A4,4X,A4,5X,4(0PE14.5))                   JFOR1910
310   WRITE (NPRT,300) J,JA,AD(K),AK(L1),A,B,C,D                        JFOR1920
320   CONTINUE                                                          JFOR1930
      ITAG=1                                                            JFOR1940
330   LINE=LINE+1                                                       JFOR1950
      IF (LINE.LE.NL) GO TO 360                                         JFOR1960
      IF (IPRINT.EQ.0) GO TO 360                                        JFOR1970
      CALL PAGE                                                         JFOR1980
      WRITE (NPRT,220)                                                  JFOR1990
      IF (IIUNIT.EQ.0.OR.IIUNIT.EQ.1) GO TO 340                         JFOR2000
      WRITE (NPRT,230)                                                  JFOR2010
      GO TO 350                                                         JFOR2020
340   WRITE (NPRT,240)                                                  JFOR2030
350   WRITE (NPRT,260)                                                  JFOR2040
      LINE=LINE+5                                                       JFOR2050
360   GO TO (370,400,420,440,460,480), ITAG                             JFOR2060
370   LL=LL+4                                                           JFOR2070
      MM=MM+5                                                           JFOR2080
```

```
C
C     OUTPUT ERROR MESSAGES
C
380   IF (IERROR(I,1).EQ.0) GO TO 400
      PRINT 390, J
390   FORMAT (1H ,23H*** JOINT NUMBER INPUT,,I3,34H, HAS NOT BEEN DEFINE
     1D (JFOR). ***)
      ITAG=2
      GO TO 330
400   IF (IERROR(I,2).EQ.0) GO TO 420
410   FORMAT (1H ,33H*** THE DISTRIBUTED LOAD OF JOINT,I4,39H HAS NO LOA
     1DED JOINT GIVEN (JFOR). *** )
      PRINT 410, J
      ITAG=3
      GO TO 330
420   IF (IERROR(I,3).EQ.0) GO TO 440
430   FORMAT (1H ,43H*** REFERENCE FUNCTION CODE INPUT FOR JOINT,I3,27H
     1IS NOT DEFINED (JFOR). ***)
      PRINT 430, J
      ITAG=4
      GO TO 330
440   IF (IERROR(I,4).EQ.0) GO TO 460
450   FORMAT (1H ,23H* TIME PERIOD FOR JOINT,I4,49H IS IRRELEVANT AND HA
     1S BEEN IGNORED (JFOR).*      )
      IREC=IREC+1
      PRINT 450, J
      ITAG=5
      GO TO 330
460   IF (IERROR(I,5).EQ.0) GO TO 480
470   FORMAT (1H ,78H*** POSITIVE TIME PERIOD MUST BE INPUT FOR REFERENC
```

```
     1E FUNCTION INPUT FOR JOINT ,I3,12H (JFOR). ***)                  JFOR2390
      IERR = IERR + 1                                                  JFOR2400
      PRINT 470, J                                                     JFOR2410
      ITAG=6                                                           JFOR2420
      GO TO 330                                                        JFOR2430
480   IF (I.LT.NFF) GO TO 280                                          JFOR2440
C                                                                      JFOR2450
C     CHECK FOR MULTIPLE OUTPUT                                        JFOR2460
C                                                                      JFOR2470
      IF (IIUNIT.EQ.0.OR.IIUNIT.EQ.2) GO TO 500                        JFOR2480
      IF (IERR.GT.IIERR) GO TO 500                                     JFOR2490
      IF (IPRINT.EQ.0) GO TO 500                                       JFOR2500
      XEGSIF = 1.E0                                                    JFOR2510
      IF(IIUNIT.EQ.1) IIUNIT = 2                                       JFOR2520
      IF(IIUNIT.EQ.3) IIUNIT = 0                                       JFOR2530
      LINE=NL+1                                                        JFOR2540
      GO TO 190                                                        JFOR2550
C                                                                      JFOR2560
C     CHECK FOR TERMINATION DUE TO LACK OF STORAGE                     JFOR2570
C                                                                      JFOR2530
500   IF (JERROR.EQ.0) GO TO 520                                       JFOR2590
510   FORMAT (1H ,71H***INSUFFICIENT STORAGE AVAILABLE FOR JOINT FORCINGJFOR2600
     1 DATA. (JFOR).***    )                                           JFOR2610
      PRINT 510                                                        JFOR2620
      GO TO 540                                                        JFOR2630
520   IF (KERROR.EQ.0) GO TO 540                                       JFOR2640
530   FORMAT (1H ,81H***INSUFFICIENT STORAGE AVAILABLE FOR JOINT FORCE IJFOR2650
     1DENTIFICATION DATA (JFCR).*** )                                  JFOR2660
      PRINT 530                                                        JFOR2670
540   RETURN                                                           JFOR2680
```

```
END                                                             JFOR2690
```

```
CLEAF    0  10
      SUBROUTINE LEAF (M,UR,IFLAG)                                      LEAF   0
C                                                                       LEAF  10
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),       LEAF  20
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),LEAF  30
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)                      LEAF  40
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), LEAF  50
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      LEAF  60
      INTEGER HEAD,DHEAD                                                LEAF  70
      COMMON /MAINBK/IDUM1(42),NPRT,IDUM2(6)                            LEAF  80
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),LEAF  90
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),    LEAF 100
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),LEAF 110
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                LEAF 120
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45), LEAF 130
     5    YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)              LEAF 140
      COMMON/STRNBK/SRP(4),SRQ(4),UX,UY,UZ,XLEN,AREA,ZZI,IMAT           LEAF 150
C                                                                       LEAF 160
C     THIS SUBROUTINE GENERATES STRESS AND ENERGY CALCULATIONS FOR A    LEAF 170
C     LEAF SPRING MEMBER(M).  THE CALCULATIONS ARE CONTROLLED           LEAF 180
C     BY (IFLAG),WHERE                                                  LEAF 190
C     IFLAG=1, INDICATES THAT THE RECOVERABLE STRAIN ENERGY(UR)         LEAF 200
C              IS REQUIRED.                                             LEAF 210
C     IFLAG=3, INDICATES THAT STRESS RESULTS ARE REQUIRED FOR PRINT-    LEAF 220
C              OUT AND FOR UPDATING THE STRESS HISTORY DATA BANK.       LEAF 230
C                                                                       LEAF 240
C INITIALIZE.                                                           LEAF 250
C                                                                       LEAF 260
      UR=0.E0                                                           LEAF 270
      ILS=IABS(M)                                                       LEAF 280
```

```
      IF (SPRING(5,ILS).LE.0.E0) GO TO 999                              LEAF 290
      CALL DEFO(M)                                                      LEAF 300
C                                                                       LEAF 310
C FIND LEAF GENERALIZED FORCES AT END *S*.                              LEAF 320
C                                                                       LEAF 330
      SRQ(1)=SPRING(4,ILS)*UX                                           LEAF 340
      SRQ(2)=SPRING(2,ILS)*UY+SPRING(3,ILS)*UZ                          LEAF 350
      SRQ(3)=SPRING(1,ILS)*UY+SPRING(2,ILS)*UZ                          LEAF 360
C                                                                       LEAF 370
C FIND LEAF STRAIN ENERGY.                                              LEAF 380
C                                                                       LEAF 390
      UR=.5E0*(SRQ(1)*UX+SRQ(2)*UZ+SRQ(3)*UY)                           LEAF 400
      IF(UR.LT.TINY) UR=0.E0                                            LEAF 410
      IF (IFLAG.NE.3) GO TO 999                                         LEAF 420
      IF (UR.LE.SPRING(5,ILS)) GO TO 999                                LEAF 430
      SPRING(5,ILS)=0.E0                                                LEAF 440
101   FORMAT(1H ,15H **LEAF SPRING ,I3,32H FAILED AT THIS LOADING.(LEAF)LEAF 450
     1**)                                                               LEAF 460
      WRITE (NPRT,101) ILS                                              LEAF 470
999   RETURN                                                            LEAF 480
      END                                                               LEAF 490
```

```
CLINK    0  10
      SUBROUTINE LINK                                                   LINK   0
C                                                                       LINK  10
C     PRINTS ELEMENT DATA OF THE LINE ELEMENT                           LINK  20
C                                                                       LINK  30
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),       LINK  40
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),LINK  50
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)                      LINK  60
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9), LINK  70
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)LINK  80
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), LINK  90
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),      LINK 100
     1    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)     LINK 110
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), LINK 120
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      LINK 130
      INTEGER HEAD,DHEAD                                                LINK 140
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,LINK 150
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,LINK 160
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, LINK 170
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  LINK 180
     4    NTIMES,NVEL,IINITD                                            LINK 190
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),LINK 200
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),    LINK 210
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),LINK 220
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                LINK 230
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45), LINK 240
     5    YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)              LINK 250
      COMMON/SCALE/EGSIF,EGSIL                                          LINK 260
      DIMENSION DU(6),DS(4),DR(15),DQ(4)                                LINK 270
      DATA DU/4HELAS,4HTIC ,4HCHAN,4HGES ,4H INE,4HLAS /               LINK 280
```

```
      DATA OS/4HIGNO,4HRE  ,4HSTOP,4H     /                            LINK 290
      DATA DR/4HR/C-,4HSTIR,4HRUPS,4H  R/,4HC-TI,4HED  ,4H WF ,4HIN R,  LINK 300
     14H/C  ,4H  WF,4H BEA,4HM   ,4HLEAF,4H SPR,4HING /                 LINK 310
      DATA DQ/4HSTIR,4HRUPS ,4H  TI,4HES  /                             LINK 320
      NPAS=1                                                            LINK 330
      ICURR=1                                                           LINK 340
      IF (IUNITS.GT.1) ICURR=2                                          LINK 350
      IF (IERR.NE.0) CALL PAGE                                          LINK 360
 20   FORMAT (1H0,20X,25HELEMENT CONTROL VARIABLES/)                    LINK 370
      WRITE (NPRT,20)                                                   LINK 380
 30   FORMAT (1H ,60H ELEMENT  JOINTS  ELEMENT TYPE  CONCRETES  BEHAVIOLINK 390
     1R  SHEAR /)                                                       LINK 400
      WRITE (NPRT,30)                                                   LINK 410
      LINE=LINE+6                                                       LINK 420
      DO 60 I=1,NM                                                      LINK 430
      IF (LINE.LT.NL) GO TO 50                                          LINK 440
      CALL PAGE                                                         LINK 450
      WRITE (NPRT,30)                                                   LINK 460
 40   FORMAT (1H ,2X,I3,4X,I3,1H-,I3,3X,3A4,2X,A4,1X,A4,2X,2A4,2X,2A4)  LINK 470
 50   IR=3*(MTYPE(I)-1)+1                                               LINK 480
      IS=2*(MSHEAR(I))+1                                                LINK 490
      IA=MCODE(I)                                                       LINK 500
      IF (MTYPE(I).EQ.4) IA=9                                           LINK 510
      IB=MATR(I)                                                        LINK 520
      IF (MTYPE(I).EQ.4) IB=MATW(I)                                     LINK 530
      IT=2*(MSTAT(I)-1)+1                                               LINK 540
      WRITE (NPRT,40) I,IP(I),IQ(I),DR(IR),DR(IR+1),DR(IR+2),NAME(IA),NALINK 550
     1ME(IB),DU(IT),DU(IT+1),DS(IS),DS(IS+1)                            LINK 560
 60   LINE=LINE+1                                                       LINK 570
 10   NWF = 0                                                           LINK 580
```

```
      DO 65 J=1,NM                                                      LINK 590
      IF (MTYPE(J).NE.4) GO TO 65                                       LINK 600
      NWF=NWF+1                                                         LINK 610
65    CONTINUE                                                          LINK 620
      CALL PAGE                                                         LINK 630
      IF(NM.EQ.NWF) GO TO 403                                           LINK 640
70    FORMAT (1H ,45X,32H BASIC ELEMENT GROSS DIMENSIONS /)             LINK 650
      WRITE (NPRT,70)                                                   LINK 660
80    FORMAT (116H  ELEMENT   JOINT     JOINT SPAN AT   JOINT SPAN AT    EFLINK 670
     1FECTIVE     SECTION        SECTION        CONFINED       CONFINED)  LINK 680
      WRITE (NPRT,80)                                                   LINK 690
90    FORMAT (118H  NUMBER   NUMBERS  1ST JOINT, IN. 2ND JOINT, IN.   LELINK 700
     1NGTH,IN.     WIDTH, IN.     HEIGHT, IN.   DEPTH, IN.     WIDTH, IN./)LINK 710
      IF (ICURR.EQ.1) WRITE (NPRT,90)                                   LINK 720
100   FORMAT (118H  NUMBER   NUMBERS  1ST JOINT, M.  2ND JOINT, M.    LELINK 730
     1NGTH, M.    WIDTH, M       HEIGHT, M      DEPTH, M       WIDTH, M  /)LINK 740
      IF (ICURR.EQ.2) WRITE (NPRT,100)                                  LINK 750
      LINE=LINE+6                                                       LINK 760
      DO 130 I=1,NM                                                     LINK 770
      XEND=XL(I)-XBEGM(I)-EFLM(I)                                       LINK 780
      IF (LINE.LE.NL) GO TO 120                                         LINK 790
      CALL PAGE                                                         LINK 800
      WRITE (NPRT,80)                                                   LINK 810
      IF (ICURR.EQ.1) WRITE (NPRT,90)                                   LINK 820
      IF (ICURR.EQ.2) WRITE (NPRT,100)                                  LINK 830
110   FORMAT (1H ,2XI3,4X,I3,1H-,I3,2X,0PE12.5,3X,0PE12.5,2X,3X,0PE12.5,LINK 840
     12X,0PE12.5,2X,0PE12.5,2X,0PE12.5,2X,0PE12.5)                      LINK 850
120   IF(MTYPE(I).EQ.4) GO TO 130                                       LINK 860
      WRITE (NPRT,110) I,IP(I),IQ(I),XBEGM(I),XEND,EFLM(I),BMEM(I),HMEM(LINK 870
     1I),DPP(I),BPP(I)                                                  LINK 880
```

```
      LINE=LINE+1                                                       LINK 890
130    CONTINUE                                                         LINK 900
      CALL PAGE                                                         LINK 910
140   FORMAT (20X,28H ELEMENT CROSS SECTION DATA /)                     LINK 920
      WRITE (NPRT,140)                                                  LINK 930
150   FORMAT (1H ,69H ELEMENT  JOINT    DEPTH- TOP TO    DEPTH- TOP TO  LINK 940
     1  DEPTH- TOP TO    )                                              LINK 950
      WRITE (NPRT,150)                                                  LINK 960
160   FORMAT (1H ,69H NUMBER   NUMBERS  LOWER REBAR,IN.  UPPER REBAR,IN.LINK 970
     1  REF. AXIS, IN.  /)                                              LINK 980
      IF (ICURR.EQ.1) WRITE (NPRT,160)                                  LINK 990
170   FORMAT (1H ,69H NUMBER   NUMBERS  LOWER REBAR, M   UPPER REBAR, M LINK1000
     1  REF. AXIS, M    /)                                              LINK1010
      IF (ICURR.EQ.2) WRITE (NPRT,170)                                  LINK1020
      LINE=LINE+6                                                       LINK1030
      DO 200 I=1,NM                                                     LINK1040
      IF (LINE.LT.NL) GO TO 190                                         LINK1050
      CALL PAGE                                                         LINK1060
      WRITE (NPRT,150)                                                  LINK1070
      IF (ICURR.EQ.1) WRITE (NPRT,160)                                  LINK1080
      IF (ICURR.EQ.2) WRITE (NPRT,170)                                  LINK1090
180   FORMAT (1H ,2X,I3,4X,I3,1H-,I3,3X,0PE12.5,5X,0PE12.5,5X,          LINK1100
     10PE12.5,4X,0PE12.5)                                               LINK1110
190    IF(MTYPE(I).EQ.4) GO TO 200                                      LINK1120
      WRITE (NPRT,180) I,IP(I),IQ(I),D(I),DP(I),HTOP(I)                 LINK1130
      LINE=LINE+1                                                       LINK1140
200    CONTINUE                                                         LINK1150
C                                                                       LINK1160
C     OUTPUT REINFORCEMENT DATA, FIRST, LONGITUDINAL,                   LINK1170
      IPST=0                                                            LINK1180
```

```
      DO 280 J=1,NM                                                     LINK1190
      NGP=NGRP(J)                                                       LINK1200
      IF (NGP.EQ.0) GO TO 280                                           LINK1210
      IF(IPST.NE.0)GO TO 246                                            LINK1220
      IPST=1                                                            LINK1230
      CALL PAGE                                                         LINK1240
210   FORMAT (1H ,36X,34H LONGITUDINAL REINFORCEMENT GROUPS/)           LINK1250
      WRITE (NPRT,210)                                                  LINK1260
220   FORMAT (112H BASE      JOINT    GROUP    SPAN-1ST JOINT  SPAN-REBARLINK1270
     1 TO EFFECTIVE      REBAR          DEPTH- REBAR TO  MAT+L )        LINK1280
      WRITE (NPRT,220)                                                  LINK1290
230   FORMAT (112H ELEMENT  NUMBERS  NUMBER  TO REBAR, IN.   2ND JOINT,LINK1300
     1IN.  REBAR SPAN,IN.  AREA,IN.**2   CENTROID, IN.            /)    LINK1310
      IF (ICURR.EC.1) WRITE (NPRT,230)                                  LINK1320
240   FORMAT (112H ELEMENT  NUMBERS  NUMBER  TO REBAR, M     2ND JOINT,LINK1330
     1 M   REBAR SPAN, M   AREA, M**2    CENTROID, M        NAME  /)    LINK1340
      IF (ICURR.EQ.2) WRITE (NPRT,240)                                  LINK1350
      LINE=LINE+6                                                       LINK1360
 246  DO 270 I=1,NGP                                                    LINK1370
      IA=MBAR(I,J)                                                      LINK1380
      XEND=XL(J)-XBEG(I,J)-EFFL(I,J)                                    LINK1390
      IF (LINE.LE.NL) GO TO 260                                         LINK1400
      CALL PAGE                                                         LINK1410
      WRITE (NPRT,220)                                                  LINK1420
      IF (ICURR.EQ.1) WRITE (NPRT,230)                                  LINK1430
      IF (ICURR.EQ.2) WRITE (NPRT,240)                                  LINK1440
250   FORMAT (1H ,3X,I3,4X,I3,1H-,I3,3X,I3,4X,0PE12.5,4X,0PE12.5,3X,0PE1LINK1450
     12.5,3X,0PE12.5,3X,0PE12.5,5X,A4)                                  LINK1460
260   WRITE (NPRT,250) J,IP(J),IQ(J),I,XBEG(I,J),XEND,EFFL(I,J),AGRP(I,JLINK1470
     1),YBAR(I,J),NAME(IA)                                              LINK1480
```

```
270   CONTINUE                                                           LINK1490
280   LINE=LINE+1                                                        LINK1500
C                                                                        LINK1510
C     THEN LATERAL.                                                      LINK1520
      IPST=0                                                             LINK1530
      DO 390 J=1,NM                                                      LINK1540
      IB=MTIES(J)                                                        LINK1550
      NLAT=NTIES(J)                                                      LINK1560
      IF (NPAS.EQ.1) YLDS(J) = YLDS(J)/(EGSIF/EGSIL**2)                  LINK1570
      IF (NLAT.EQ.0) GO TO 390                                           LINK1580
      IF(IPST.NE.0)GO TO 336                                             LINK1590
      IPST=1                                                             LINK1600
      CALL PAGE                                                          LINK1610
290   FORMAT (1H ,43X,32H LATERAL REINFORCEMENT GROUPS   /)              LINK1620
      WRITE (NPRT,290)                                                   LINK1630
300   FORMAT (118H  BASE     JOINT     GROUP    REBAR     MAT*L  REBAR   LINK1640
     1  SPACING OF     NO. OF START OF       YIELD STRESS  CONFINEMENT)  LINK1650
      WRITE (NPRT,300)                                                   LINK1660
310   FORMAT (119H  ELEMENT  NUMBERS  NUMBER  T YPE       NAME   AREA,IN.**LINK1670
     12   REBARS,  IN.  SPACES  GROUP, IN.    LB/IN.**2       FACTOR *PDP*/LINK1680
     2)                                                                  LINK1690
      IF (ICURR.EQ.1) WRITE (NPRT,310)                                   LINK1700
320   FORMAT (119H  ELEMENT  NUMBER   NUMBER  TYPE        NAME   AREA, M**2LINK1710
     1    REBARS, M.    SPACES  GROUP, M.       N/M**2        FACTOR *PDP*/LINK1720
     2)                                                                  LINK1730
      IF (ICURR.EQ.2) WRITE (NPRT,320)                                   LINK1740
      LINE=LINE+6                                                        LINK1750
 336  IF (NSPAC(6,J).EQ.-1) GO TO 360                                    LINK1760
      DO 350 I=1,NLAT                                                    LINK1770
      IF (LINE.LE.NL) GO TO 340                                          LINK1780
```

```
      CALL PAGE                                                         LINK1790
      WRITE (NPRT,300)                                                  LINK1800
      IF (ICURR.EQ.1) WRITE (NPRT,310)                                  LINK1810
      IF (ICURR.EQ.2) WRITE (NPRT,320)                                  LINK1820
  330 FORMAT (1H ,2X,I3,4X,I3,1H-,I3,3X,I3,5X,A4,A4,2X,A4,1X,0PE12.5,2X,LINK1830
     10PE12.5,3X,I3,4X,0PE11.4,2X,0PE12.5,2X,0PE12.5)                   LINK1840
  340 WRITE (NPRT,330) J,IP(J),IQ(J),I,DQ(1),DQ(2),NAME(IB),ATIES(I,J),SLINK1850
     1TIES(I,J),NSPAC(I,J),XBEGS(I,J),YLDS(J),PDP(I,J)                  LINK1860
  350 LINE=LINE+1                                                       LINK1870
      GO TO 390                                                         LINK1880
  360 DO 380 I=1,NLAT                                                   LINK1890
      IF (LINE.LE.NL) GO TO 370                                         LINK1900
      CALL PAGE                                                         LINK1910
      WRITE (NPRT,300)                                                  LINK1920
      IF (ICURR.EQ.1) WRITE (NPRT,310)                                  LINK1930
      IF (ICURR.EQ.2) WRITE (NPRT,320)                                  LINK1940
  370 WRITE (NPRT,330) J,IP(I),IQ(I),I,DQ(3),DQ(4),NAME(IB),ATIES(I,J),SLINK1950
     1TIES(I,J),NSPAC(I,J),XBEGS(I,J),YLDS(J),PDP(I,J)                  LINK1960
  380 LINE=LINE+1                                                       LINK1970
  390 CONTINUE                                                          LINK1980
C                                                                       LINK1990
C     OUTPUT MEMBER DATA ON STEEL FOR COMPOSITE SECTIONS                LINK2000
      IF (NCM.EQ.0.AND.NWF.EQ.0) GO TO 470                              LINK2010
      CALL PAGE                                                         LINK2020
      IF(NWF.NE.0) GO TO 403                                            LINK2030
  400 FORMAT (1H ,21X,46H WIDE FLANGE DIMENSIONS FOR COMPOSITE ELEMENTS/LINK2040
     1)                                                                 LINK2050
      WRITE (NPRT,400)                                                  LINK2060
      GO TO 413                                                         LINK2070
  403 WRITE (NPRT,405)                                                  LINK2080
```

```
405   FORMAT (1H ,21X,23H WIDE FLANGE DIMENSIONS/)                      LINK2090
410   FORMAT (1H ,75H BASE     JOINT    THICKNESS,    WIDTH OF      THICLINK2100
     1KNESS     DEPTH OF      )                                         LINK2110
413   WRITE (NPRT,410)                                                  LINK2120
420   FORMAT (1H ,79H ELEMENT  NUMBERS  FLANGE,IN.    FLANGE,IN.    OF WLINK2130
     1EB, IN.   BEAM, IN.         /)                                    LINK2140
      IF (ICURR.EQ.1) WRITE (NPRT,420)                                  LINK2150
430   FORMAT (1H ,68H ELEMENT  NUMBERS  FLANGE, M     FLANGE, M     OF WLINK2160
     1EB, M     BEAM, M/)                                               LINK2170
      IF (ICURR.EQ.2) WRITE (NPRT,430)                                  LINK2180
      DO 460 I=1,NM                                                     LINK2190
      IF (LINE.LE.NL) GO TO 450                                         LINK2200
      CALL PAGE                                                         LINK2210
      WRITE (NPRT,410)                                                  LINK2220
      IF (ICURR.EQ.2) WRITE (NPRT,430)                                  LINK2230
      IF (ICURR.EQ.1) WRITE (NPRT,420)                                  LINK2240
440   FORMAT (1H ,2X,I3,4X,I3,1H-,I3,2X,4(0PE12.5,2X))                  LINK2250
450     IF(MTYPE(I).LT.3) GO TO 460                                     LINK2260
      WRITE (NPRT,440) I,IP(I),IQ(I),TFWF(I),BWF(I),TWWF(I),DWF(I)      LINK2270
460   CONTINUE                                                          LINK2280
C                                                                       LINK2290
C     OUTPUT LEAF SPRING DATA.                                          LINK2300
470   IF (NLS.EQ.0) GO TO 550                                           LINK2310
      CALL PAGE                                                         LINK2320
480   FORMAT (1H ,26X,31H LEAF SPRING ELEMENT PARAMETERS/)              LINK2330
      WRITE (NPRT,480)                                                  LINK2340
490   FORMAT (1H ,101H LEAF      JOINT     DISPL. OF TIP  ROTATION OF   ROLINK2350
     1TATION OF TIP    ELONGATION OF      STORABLE ENERGY)              LINK2360
      WRITE (NPRT,490)                                                  LINK2370
500   FORMAT (1H ,101H SPRING   NUMBERS  FORCE, LB/IN.  FORCE,LB/RAD  MOLINK2380
```

```
     1MENT,IN.*LB/RAD  AXIS FORCE,LB/IN.    LIMIT,IN.-LB  /)              LINK2390
      IF (ICURR.EQ.1) WRITE (NPRT,500)                                    LINK2400
510   FORMAT (1H ,101H SPRING   NUMBERS  FORCE, N/M      FORCE,N/RAD   MOLINK2410
     1MENT, M*N/RAD    AXIS FORCE, N/M       LIMIT,M-N     /)             LINK2420
      IF (ICURR.EQ.2) WRITE (NPRT,510)                                    LINK2430
      DO 540 I=1,NLS                                                      LINK2440
      IF (LINE.LE.NL) GO TO 530                                           LINK2450
      CALL PAGE                                                           LINK2460
      WRITE (NPRT,490)                                                    LINK2470
      IF (ICURR.EQ.2) WRITE (NPRT,510)                                    LINK2480
      IF (ICURR.EQ.1) WRITE (NPRT,500)                                    LINK2490
520   FORMAT (1H ,3X,I3,4X,I3,1H-,I3,2X,0PE12.5,3X,0PE12.5,2X,0PE12.5,7XLINK2500
     1,0PE12.5,8X,0PE12.5)                                                LINK2510
530   WRITE (NPRT,520) I,IPL(I),IQL(I),(SPRING(J,I),J=1,5)                LINK2520
540   CONTINUE                                                            LINK2530
C                                                                         LINK2540
C     MAKE SI,ENGLISH CONVERSIONS.                                        LINK2550
C                                                                         LINK2560
  550 IF(IUNITS.EQ.0.OR.IUNITS.EQ.2.OR.NPAS.GT.1) RETURN                  LINK2570
      NPAS = NPAS + 1                                                     LINK2580
      IF(IUNITS.EQ.1) ICURR = 2                                           LINK2590
      IF(IUNITS.EQ.3) ICURR = 1                                           LINK2600
      DO 580 MEM=1,NM                                                     LINK2610
      XL(MEM) = XL(MEM)*EGSIL                                             LINK2620
      BMEM(MEM)=BMEM(MEM)*EGSIL                                           LINK2630
      BPP(MEM)=BPP(MEM)*EGSIL                                             LINK2640
      D(MEM)=D(MEM)*EGSIL                                                 LINK2650
      DP(MEM)=DP(MEM)*EGSIL                                               LINK2660
      DPP(MEM)=DPP(MEM)*EGSIL                                             LINK2670
      EFLM(MEM)=EFLM(MEM)*EGSIL                                           LINK2680
```

```
      HMEM(MEM)=HMEM(MEM)*EGSIL
      XBEGM(MEM)=XBEGM(MEM)*EGSIL
      YLDS(MEM) = YLDS(MEM)*EGSIF/EGSIL**2
      DWF(MEM)=DWF(MEM)*EGSIL
      TFWF(MEM)=TFWF(MEM)*EGSIL
      TWWF(MEM)=TWWF(MEM)*EGSIL
      BWF(MEM)=BWF(MEM)*EGSIL
      HTOP(MEM)=HTOP(MEM)*EGSIL
      HTWF(MEM)=HTWF(MEM)*EGSIL
C
      DO 570 I=1,6
      ATIES(I,MEM) =  ATIES(I,MEM)*EGSIL**2
      STIES(I,MEM)=STIES(I,MEM)*EGSIL
570   XBEGS(I,MEM)=XBEGS(I,MEM)*EGSIL
      DO 580 IGRP=1,10
      AGRP(IGRP,MEM)  =  AGRP(IGRP,MEM)*EGSIL**2
      EFFL(IGRP,MEM)=EFFL(IGRP,MEM)*EGSIL
      XBEG(IGRP,MEM)=XBEG(IGRP,MEM)*EGSIL
580   YBAR(IGRP,MEM)=YBAR(IGRP,MEM)*EGSIL
      DO 590 I=1,NLS
      DO 585 J=1,4
585   SPRING(J,I)=SPRING(J,I)*EGSIL/EGSIF
590   SPRING(5,I)=SPRING(5,I)*EGSIL*EGSIF
      DO 595 J=1,NJ
      X(J) = X(J)*EGSIL
595   Y(J) = Y(J)*EGSIL
      GO TO 10
      END
```

```
CLUMP    0  10
      SUBROUTINE LUMP                                                   LUMP   0
C                                                                       LUMP  10
C     THIS SUBROUTINE PRINTS OUT LUMPED MASS INPUT DATA.                LUMP  20
C                                                                       LUMP  30
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), LUMP  40
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),      LUMP  50
     1    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)     LUMP  60
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), LUMP  70
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      LUMP  80
      INTEGER HEAD,DHEAD                                                LUMP  90
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,LUMP 100
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,LUMP 110
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, LUMP 120
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  LUMP 130
     4    NTIMES,NVEL,IINITD                                            LUMP 140
      COMMON/SCALE/EGSIF,EGSIL                                          LUMP 150
C                                                                       LUMP 160
C     PRINT HEADING WITH EITHER EG OR SI DIMENSIONS.                    LUMP 170
      NPAGE= 1                                                          LUMP 180
20    CALL PAGE                                                         LUMP 190
30    FORMAT (1H ,20A4,//)                                              LUMP 200
      WRITE (NPRT,30) DHEAD                                             LUMP 210
40    FORMAT (1H ,15X,32HLUMPED MASSES AS READ FROM CARDS)              LUMP 220
      WRITE (NPRT,40)                                                   LUMP 230
50    FORMAT (1H ,14X,35HPLUS MEMBER MASSES LUMPED AT JOINTS)           LUMP 240
      WRITE (NPRT,50)                                                   LUMP 250
60    FORMAT (1H0)                                                      LUMP 260
      WRITE (NPRT,60)                                                   LUMP 270
70    FORMAT (1H ,13X,9HMASS  FOR,9X,9HMASS  FOR,8X,11HMASS MOMENT/1H ,5LUMP 280
```

```
      1HJOINT,7X,11HX-DIRECTION,7X,11HY-DIRECTION,7X,11HOF   INERTIA)
       WRITE (NPRT,70)
       LINE=LINE+9
90     FORMAT (1H ,12X,11HLB*S**2/IN.,7X,11HLB*S**2/IN.,7X,11HLB*S**2*IN.
      1/)
       IF((NPAGE.EQ.1.AND.IUNITS.LE.1).OR. (NPAGE.EQ.2.AND.
      1    (IUNITS.EQ.0.OR.IUNITS.EQ.3))) WRITE (NPRT,90)
110    FORMAT (1H ,17X,2HKG,16X,2HKG,13X,7HKG*M**2/)
       IF((NPAGE.EQ.1.AND.IUNITS.GT.1).OR. (NPAGE.EQ.2.AND.
      1    (IUNITS.EQ.1.OR.IUNITS.EQ.2))) WRITE (NPRT,110)
       LINE=LINE+2
C
C      BEGIN DO LOOP THAT PRINTS OUT NONZERO MASS PARAMETER VALUES.
       DO 240  J=1,NJ
       IF (DAS(1,J).LT.EPS.AND.DAS(2,J).LT.EPS.AND.DAS(3,J).LT.EPS) GO TO
      1 240
C
C      PRINT OUT LUMPED MASSES.
160    FORMAT (1H ,I5,1P3E18.4)
       WRITE(NPRT,160)  J,(DAS(I,J),I=1,3)
       LINE=LINE+1
C
C      CONVERT UNITS OF LUMPED MASSES.
       IF(NPAGE.EQ.2)  GO TO 200
       DAS(1,J)  =  DAS(1,J)*(EGSIF/EGSIL)
       DAS(2,J)  =  DAS(2,J)*(EGSIF/EGSIL)
       DAS(3,J)  =  DAS(3,J)*(EGSIF*EGSIL)
C
C      TEST LINE COUNTER AND REPEAT HEADING IF NECESSARY.
C
```

```
200   IF (LINE.LE.NL) GO TO 240                                      LUMP 590
      CALL PAGE                                                      LUMP 600
      WRITE (NPRT,40)                                                LUMP 610
      WRITE (NPRT,50)                                                LUMP 620
      WRITE (NPRT,60)                                                LUMP 630
      WRITE (NPRT,70)                                                LUMP 640
      LINE=LINE+6                                                    LUMP 650
      IF (NPAGE.EQ.1) GO TO 210                                      LUMP 660
      IF (IUNITS.EQ.1) GO TO 230                                     LUMP 670
      GO TO 220                                                      LUMP 680
210   IF (IUNITS.GT.1) GO TO 230                                     LUMP 690
220   WRITE (NPRT,90)                                                LUMP 700
      LINE=LINE+2                                                    LUMP 710
      GO TO 240                                                      LUMP 720
230   WRITE (NPRT,110)                                               LUMP 730
      LINE=LINE+2                                                    LUMP 740
C                                                                    LUMP 750
C     END OF DO LOOP.                                                LUMP 760
C                                                                    LUMP 770
240   CONTINUE                                                       LUMP 780
C                                                                    LUMP 790
C     IF INPUT-OUTPUT UNITS ARE MIXED, PRINT SECOND SET OF DATA.     LUMP 800
C                                                                    LUMP 810
      IF (NPAGE.EQ.2) GO TO 250                                      LUMP 820
      IF (IUNITS.EQ.0.OR.IUNITS.EQ.2) GO TO 250                      LUMP 830
      NPAGE=2                                                        LUMP 840
      GO TO 20                                                       LUMP 850
250   RETURN                                                         LUMP 860
C                                                                    LUMP 870
      END                                                            LUMP 880
```

```
CMASS    0  10
      SUBROUTINE MASS                                                   MASS   0
C                                                                       MASS  10
C     THIS SUBROUTINE READS FROM CARDS LUMPED MASSES AT THE JOINTS.     MASS  20
C                                                                       MASS  30
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),       MASS  40
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),MASS  50
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)                      MASS  60
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), MASS  70
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),      MASS  80
     1    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)     MASS  90
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), MASS 100
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      MASS 110
      INTEGER HEAD,DHEAD                                                MASS 120
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,MASS 130
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,MASS 140
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, MASS 150
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  MASS 160
     4    NTIMES,NVEL,IINITD                                            MASS 170
C                                                                       MASS 180
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  MASS 190
C     READ USER*S TITLE AND INITIALIZE JOINT AND ERROR COUNTER.         MASS 200
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  MASS 210
C                                                                       MASS 220
10    FORMAT (20A4)                                                     MASS 230
      READ (NCRD,10) DHEAD                                              MASS 240
      NMAS=0                                                            MASS 250
      JERR=0                                                            MASS 260
C                                                                       MASS 270
      DO 20 J=1,NJ                                                      MASS 280
```

```
      DO 20 I=1,3                                                       MASS 290
20    DAS(I,J)=0.E0                                                     MASS 300
C     READ JOINT NO. AND LUMPED MASSES AT THE JOINT.                    MASS 310
C                                                                       MASS 320
30    FORMAT (I5,5X,3E10.0)                                             MASS 330
40    READ (NCRD,30) J,B,C,D                                            MASS 340
C                                                                       MASS 350
C     TEST TO SEE IF LAST CARD.                                         MASS 360
C                                                                       MASS 370
      IF (J.EQ.0) GO TO 170                                             MASS 380
C                                                                       MASS 390
C     INCREMENT JOINT COUNTER AND CHECK RANGE OF JOINT NUMBER.          MASS 400
C                                                                       MASS 410
      NMAS=NMAS+1                                                       MASS 420
      IF (J.GE.1.AND.J.LE.NJ) GO TO 90                                  MASS 430
      IERR=IERR+1                                                       MASS 440
      IF (JERR.EQ.1) GO TO 80                                           MASS 450
      CALL PAGE                                                         MASS 460
50    FORMAT (1H ,20A4,//)                                              MASS 470
      WRITE (NPRT,50) DHEAD                                             MASS 480
60    FORMAT (1H ,10X,41HINPUT ERRORS IN EXTERNAL LUMPED MASS DATA//)   MASS 490
      WRITE (NPRT,60)                                                   MASS 500
      JERR=1                                                            MASS 510
70    FORMAT (14H *** JOINT NO.,I5,50H LESS THAN ONE OR GREATER THAN LARMASS 520
     1GEST JOINT NO. =,I4,12H (MASS). ***/)                             MASS 530
80    PRINT 70, J,NJ                                                    MASS 540
      GO TO 40                                                          MASS 550
90    IF (NJ.GT.NJD) GO TO 40                                           MASS 560
C                                                                       MASS 570
C                                                                       MASS 580
```

```
C     IF UNEQUAL MASSES DETECTED, PRINT WARNING AND CONTINUE.
C
      IF (ABS(B-C).LT.EPS) GO TO 130
      IF (JERR.EQ.1) GO TO 120
      CALL PAGE
      WRITE (NPRT,60)
      JERR=1
110   FORMAT (1H ,66H** WARNING. INPUT VALUES OF TRANSLATIONAL MASS PARA
     1METERS AT JOINT,I5,43H NOT EQUAL IN X AND Y DIRECTIONS (MASS). **)
120   PRINT 110, J
      IREC=IREC+1
C
C     IF MASS IS NEGATIVE, STORE POSITIVE VALUE AND CONTINUE.
C
130   IF (B.GT.-EPS.OR.C.GT.-EPS) GO TO 160
      IREC=IREC+1
      IF (JERR.EQ.1) GO TO 150
      CALL PAGE
      WRITE (NPRT,60)
      JERR=1
140   FORMAT (34H * NEGATIVE MASS DETECTED AT JOINT,I5,41H IGNORED, POSI
     1TIVE VALUE STORED (MASS). *)
150   PRINT 140,J
      IREC=IREC+1
      DAS(1,J)=ABS(B)
      DAS(2,J)=ABS(C)
      DAS(3,J)=0
      GO TO 40
160   DAS(1,J)=B
      DAS(2,J)=C
```

```
      DAS(3,J)=0                                                      MASS 890
      GO TO 40                                                        MASS 900
C                                                                     MASS 910
170   RETURN                                                          MASS 920
      END                                                             MASS 930
```

```
CMATP    0  10
      SUBROUTINE MATP                                                   MATP   0
C                                                                       MATP  10
C     THIS SUBROUTINE READS FROM CARDS,ERROR CHECKS, AND STORES INFORMATMATP  20
C     DEFINING THE MATERIAL PROPERTIES                                  MATP  30
C                                                                       MATP  40
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9), MATP  50
     1   SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)MATP  60
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), MATP  70
     1   PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB       MATP  80
      INTEGER HEAD,DHEAD,BBBB,ZZZZ,PAME                                 MATP  90
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,MATP 100
     1   IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,MATP 110
     2   NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,  MATP 120
     3   NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,   MATP 130
     4   NTIMES,NVEL,IINITD                                             MATP 140
      COMMON/SCALE/EGSIF,EGSIL                                          MATP 150
      DIMENSION IERROR(9,16),STSK(8),TYPA(5,3)                          MATP 160
C                                                                       MATP 170
      DATA TYPA/4HUNCO,4HNFIN,4HED C,4HONCR,4HETE ,                     MATP 180
     1          4HCONF,4HINED,4H CON,4HCRET,4HE    ,                    MATP 190
     2          4HREIN,4HFORC,4HING ,4HSTEE,4HL    /                    MATP 200
      DATA BBBB/4H    /,ZZZZ/4H0000/,UA/1HU/,CS/1HC/,FT/2H$$/           MATP 210
C                                                                       MATP 220
C     INITIALIZE PARAMETERS USED BY SUBROUTINE                          MATP 230
C                                                                       MATP 240
      J=0                                                               MATP 250
      NMAT=0                                                            MATP 260
      IIERR=IERR                                                        MATP 270
      DO 10 K=1,16                                                      MATP 280
```

```
      DO 10 M=1,9                                                       MATP 290
10    IERROR(M,K)=0                                                     MATP 300
      EGSIS = EGSIF/EGSIL**2                                            MATP 310
      EGSID = 1.E0                                                      MATP 320
      EGSISS = 1.E0                                                     MATP 330
C                                                                       MATP 340
C     CHANGE UNITS OF STORED STRESS-STRAIN CURVES.                      MATP 350
      IF(IUNITS.EQ.0.OR.IUNITS.EQ.3) GO TO 21                           MATP 360
      EGSISS = 6894.7572E0                                              MATP 370
      EGSID = 271447.1375E0                                             MATP 380
      DO 20 I=1,6                                                       MATP 390
      DO 20 K=1,9                                                       MATP 400
   20 ST(K,I) = ST(K,I)*EGSISS                                          MATP 410
   21 CONTINUE                                                          MATP 420
C                                                                       MATP 430
C     READ DATA BLOCK TITLE CARD                                        MATP 440
      READ (NCRD,30) DHEAD                                              MATP 450
30    FORMAT (20A4)                                                     MATP 460
C                                                                       MATP 470
C     READ FIRST DATA CARD                                              MATP 480
C                                                                       MATP 490
40    FORMAT (A4,1X,A1,4X,7E10.0)                                       MATP 500
50    READ (NCRD,40) PAME,TC,A,B,C,D,E,F                                MATP 510
C     PAME = MATERIAL NAME                                              MATP 520
C     TC = TYPE OF MATERIAL-U=0=UNCONF. CONC., C=1=CONFINED CONC.,2=STEEMATP 530
C     A = CRUSHING (CONCRETE) OR YIELD (STEEL) STRENGTH                 MATP 540
C     B = POISSON*S RATIO, LINEAR ELASTIC                               MATP 550
C     C = SHEAR MODULUS, LINEAR ELASTIC                                 MATP 560
C     D = DENSITY                                                       MATP 570
C     E = UNLOADING CURVE DATA (K FOR CONCRETE)                         MATP 580
```

```
C      F = UNLOADING CURVE DATA                                          MATP 590
C      QAME = MATERIAL IDENTIFICATION DATA (NAME)                        MATP 600
C                                                                        MATP 610
C      CHECK FOR LAST CARD INPUT TO DATA BLOCK                           MATP 620
       IF ((PAME.EQ.BBBB).OR.(PAME.EQ.ZZZZ)) GO TO 570                   MATP 630
C                                                                        MATP 640
C      INCREMENT MATERIAL COUNTER                                        MATP 650
       J=J+1                                                             MATP 660
       ICODE(J)=2                                                        MATP 670
       IF (TC.EQ.UA) ICODE(J)=0                                          MATP 680
       IF (TC.EQ.CS) ICODE(J)=1                                          MATP 690
       IC=ICODE(J)+1                                                     MATP 700
C                                                                        MATP 710
C      CHECK FOR AVAILABILITY OF STORAGE                                 MATP 720
       IF (J.LE.NMATD) GO TO 70                                          MATP 730
       J=NMATD                                                           MATP 740
       IERROR(J,1)=1                                                     MATP 750
       IERR=IERR+1                                                       MATP 760
C                                                                        MATP 770
C      IF STORAGE NOT AVAILABLE, READ REMAINING DATA CARDS FOR MATERIAL  MATP 780
       READ (NCRD,60) AZ,AZ                                              MATP 790
60     FORMAT (A4/A4)                                                    MATP 800
       GO TO 560                                                         MATP 810
C                                                                        MATP 820
C      STORE DATA FROM FIRST CARD                                        MATP 830
70     NAME(J)=PAME                                                      MATP 840
       PR(J)=B                                                           MATP 850
C                                                                        MATP 860
C      READ REMAINING DATA CARDS FOR MATERIAL                            MATP 870
C                                                                        MATP 880
```

```
      READ (NCRD,40) PAME,TC,(STS(I,J),STN(I,J),I=1,3),STS(4,J)          MATP 890
      IF ((PAME.NE.NAME(J)).AND.(PAME.NE.3BBB).AND.(PAME.NE.ZZZZ)) GO TOMATP 900
     1 90                                                                 MATP 910
      GO TO 100                                                           MATP 920
80    FORMAT (53H ***MATERIAL DATA CARDS ARE OUT-OF-SORT FOR MATERIAL ,AMATP 930
     14,12H (MATP).*** )                                                  MATP 940
90    PRINT 80, NAME(J)                                                   MATP 950
      IERR=IERR+1                                                         MATP 960
      READ (NCRD,40) PAME,TC,STN(4,J),(STS(I,J),STN(I,J),I=5,7)           MATP 970
      GO TO 50                                                            MATP 980
100   READ (NCRD,40) PAME,TC,STN(4,J),(STS(I,J),STN(I,J),I=5,7)           MATP 990
      IF ((PAME.NE.NAME(J)).AND.(PAME.NE.BBBB).AND.(PAME.NE.ZZZZ)) GO TOMATP1000
     1 110                                                                MATP1010
      GO TO 120                                                           MATP1020
110   PRINT 80, NAME(J)                                                   MATP1030
      IERR=IERR+1                                                         MATP1040
      GO TO 50                                                            MATP1050
C                                                                         MATP1060
C     CHECK FOR MONOTONICITY OF STRAINS AND INTERCHANGE CARD DATA IF      MATP1070
C     NECESSARY                                                           MATP1080
120   ITIME=0                                                             MATP1090
      DO 170 I=2,7                                                        MATP1100
      IF (ABS(STN(I-1,J)).GT.ABS(STN(I,J))) GO TO 140                     MATP1110
      GO TO 170                                                           MATP1120
140   IF (ITIME.GT.0) GO TO 160                                           MATP1130
      ITIME=1                                                             MATP1140
      DO 150 I1=1,7                                                       MATP1150
150   STSK(I1)=STS(I1,J)                                                  MATP1160
      STS(1,J)=STN(4,J)                                                   MATP1170
      STS(2,J)=STN(5,J)                                                   MATP1180
```

```
      STS(3,J)=STN(6,J)                                                 MATP1190
      STS(4,J)=STN(7,J)                                                 MATP1200
      STS(5,J)=STN(1,J)                                                 MATP1210
      STS(6,J)=STN(2,J)                                                 MATP1220
      STS(7,J)=STN(3,J)                                                 MATP1230
      STN(1,J)=STSK(5)                                                  MATP1240
      STN(2,J)=STSK(6)                                                  MATP1250
      STN(3,J)=STSK(7)                                                  MATP1260
      STN(4,J)=STSK(1)                                                  MATP1270
      STN(5,J)=STSK(2)                                                  MATP1280
      STN(6,J)=STSK(3)                                                  MATP1290
      STN(7,J)=STSK(4)                                                  MATP1300
      GO TO 170                                                         MATP1310
160   IERROR(J,15)=1                                                    MATP1320
      IERR=IERR+1                                                       MATP1330
      GO TO 180                                                         MATP1340
170   CONTINUE                                                          MATP1350
C                                                                       MATP1360
180   IF (A.EQ.0.E0)  A = STS(2,J)                                      MATP1370
C     STORE INPUT DATA AND CONVERT UNITS.                               MATP1380
      FCFY(J) = A*EGSIS                                                 MATP1390
      DENS(J) = D*(EGSIS/EGSIL)                                         MATP1400
      G(J) = C*EGSIS                                                    MATP1410
      E = E*EGSIS                                                       MATP1420
      DO 200 I=1,8                                                      MATP1430
  200 STS(I,J) = STS(I,J)* EGSIS                                        MATP1440
      IF (IERROR(J,3).NE.0.OR.IERROR(J,4).NE.0) GO TO 560               MATP1450
      IF (ICODE(J).EQ.2) GO TO 225                                      MATP1460
C                                                                       MATP1470
C     CHECK FOR REVERSE LOADING STRESS-STRAIN POINT SPECIFIED FOR CONCREMATP1480
```

```
      IF (E.NE.0.E0.OR.F.NE.0.E0) GO TO 220                            MATP1490
      E=.25E0                                                           MATP1500
      F=0.E0                                                            MATP1510
220   UNLK(J)=E                                                         MATP1520
      GO TO 227                                                         MATP1530
225   UNLK(J)=1.E0                                                      MATP1540
227   STN(8,J)=F                                                        MATP1550
      STS(8,J)=E                                                        MATP1560
C                                                                       MATP1570
C     CHECK THAT INPUT VALUE OF POISSON*S RATIO IS WITHIN LIMITS        MATP1580
      IF (PR(J).GE.0.E0.AND.PR(J).LT.0.5E0) GO TO 230                   MATP1590
      IERROR(J,7)=1                                                     MATP1600
      IREC = IREC + 1                                                   MATP1610
C                                                                       MATP1620
C     CHECK THAT INPUT VALUE OF SHEAR MODULUS IS WITHIN LIMITS.         MATP1630
  230 IF(G(J).GE.0.AND.G(J).LT.1.5E7*EGSISS) GO TO 240                 MATP1640
      IERROR(J,8)=1                                                     MATP1650
      IERR=IERR+1                                                       MATP1660
C                                                                       MATP1670
C     CHECK THAT INPUT VALUE OF MATERIAL DENSITY IS WITHIN LIMITS       MATP1680
C                                                                       MATP1690
240   IF(DENS(J).GE.0.E0.AND.DENS(J).LE.0.5E0*EGSID) GO TO 250          MATP1700
      IERROR(J,9)=1                                                     MATP1710
      IREC = IREC + 1                                                   MATP1720
C                                                                       MATP1730
C     CHECK THAT INPUT VALUE OF UNLOADING CURVE CONSTANT IS WITHIN LIMITMATP1740
 250  IF(ICODE(J).EQ.2)GO TO 256                                        MATP1750
 254  IF (STS(8,J).GT.0.E0.AND.STS(8,J).LE.1.E0) GO TO 260              MATP1760
      IERROR(J,10)=1                                                    MATP1770
      IERR=IERR+1                                                       MATP1780
```

```
      GO TO 260                                                         MATP1790
  256 IF(STS(8,J).LT.0.E0.OR.STN(8,J).LT.0.E0)GO TO 254                 MATP1800
  260 DO 270 I=1,7                                                      MATP1810
      IF (STS(I,J).NE.0.E0.OR.STN(I,J).NE.0.E0) GO TO 440               MATP1820
  270 CONTINUE                                                          MATP1830
      IF (IC.EQ.3) GO TO 340                                            MATP1840
C     ******************************************************************MATP1850
C     GENERATE STRESS-STRAIN CURVE FOR CONCRETE (CONFINED,UNCONFINED).  MATP1860
C     ******************************************************************MATP1870
C                                                                       MATP1880
C     CHECK THAT INPUT VALUE OF  CRUSHING STRENGTH IS WITHIN LIMITS     MATP1890
      IF(FCFY(J).GE.2.5E3*EGSISS.AND.FCFY(J).LE.8.E3*EGSISS) GO TO 300  MATP1900
      IERROR(J,11)=1                                                    MATP1910
      IERR=IERR+1                                                       MATP1920
      DO 290 I=1,7                                                      MATP1930
  290 SLOPE(I,J)=99999.E0                                               MATP1940
      EPSU(J)=99999.E0                                                  MATP1950
      SLOPE(8,J)=99999.E0                                               MATP1960
      ET(J)=99999.E0                                                    MATP1970
      EC(J)=99999.E0                                                    MATP1980
      GO TO 560                                                         MATP1990
  300 STS(1,J)=0.E0                                                     MATP2000
      STS(2,J) = -.5E0*FCFY(J)                                          MATP2010
      STS(3,J)=-.85E0*FCFY(J)                                           MATP2020
      STS(4,J)=-1.0E0*FCFY(J)                                           MATP2030
      STS(7,J)=-2.E-1*FCFY(J)                                           MATP2040
      STN(1,J)=0.0E0                                                    MATP2050
      STN(2,J)=-.586E-3                                                 MATP2060
      STN(3,J)=-1.225E-3                                                MATP2070
      STN(4,J)=-2.0E-3                                                  MATP2080
```

```
      IF (IC.EQ.2) GO TO 310                                            MATP2090
C                                                                       MATP2100
C     COMPUTE REMAINING POINTS  FOR UNCONFINED CONCRETE.                MATP2110
      STS(5,J) = -.5E0*FCFY(J)                                          MATP2120
      STS(6,J) = -.35E0*FCFY(J)                                         MATP2130
      EU=(3.0E0+2.0E-3*FCFY(J))/(FCFY(J)-1.0E3)                         MATP2140
      SLP=(EU-2.0E-3)/(.5E0*FCFY(J))                                    MATP2150
      STN(7,J) = STN(4,J) + SLP*(-.8E0*FCFY(J))                         MATP2160
      STN(5,J)=-EU                                                      MATP2170
      STN(6,J)=(STN(5,J)+STN(7,J))*.5E0                                 MATP2180
      EPSU(J)=STN(4,J)                                                  MATP2190
      GO TO 470                                                         MATP2200
C                                                                       MATP2210
C     COMPUTE REMAINING POINTS FOR CONFINED CONCRETE.                   MATP2220
310   STS(5,J) = 0.E0                                                   MATP2230
      STS(6,J) = -.2E0*FCFY(J)                                          MATP2240
      STN(7,J) = -0.3E0                                                 MATP2250
C     REMAINING POINTS (DETERMINED BY TIE SPACING) COMPUTED IN *CONC* . MATP2260
      GO TO 470                                                         MATP2270
C     ******************************************************************MATP2280
C     GENERATE STRESS-STRAIN CURVE FOR STEEL                            MATP2290
C     ******************************************************************MATP2300
C                                                                       MATP2310
C     CHECK FOR REVERSE LOADING POINT INPUT FOR INTERNALLY              MATP2320
C     GENERATED STRESS-STRAIN CURVE.                                    MATP2330
340   NST = 9                                                           MATP2340
      IF (STN(8,J).EQ.0.E0.AND.STS(8,J).EQ.0.E0) GO TO 360              MATP2350
      NST=8                                                             MATP2360
C                                                                       MATP2370
C     CHECK FOR ALLOWABLE VALUE OF YIELD STRESS                         MATP2380
```

```
C                                                                  MATP2390
360   FA=ABS(FCFY(J))                                              MATP2400
      DO 380 I=1,7                                                 MATP2410
      STN(I,J)=99999.E0                                            MATP2420
      STS(I,J)=99999.E0                                            MATP2430
380   SLOPE(I,J)=99999.E0                                          MATP2440
      SLOPE(8,J)=99999.E0                                          MATP2450
      EPSU(J)=99999.E0                                             MATP2460
      ET(J)=99999.E0                                               MATP2470
      EC(J)=99999.E0                                               MATP2480
C                                                                  MATP2490
C     GENERATE STRESS-STRAIN CURVE                                 MATP2500
C                                                                  MATP2510
C     FIND INTERPOLATION RANGE.                                    MATP2520
      DO 390 I=1,6                                                 MATP2530
      IF (FCFY(J).GE.ST(1,I)) GO TO 390                            MATP2540
      I1=I-1                                                       MATP2550
      GO TO 400                                                    MATP2560
390   CONTINUE                                                     MATP2570
      I1 = I - 1                                                   MATP2580
      IF (FCFY(J).GT.ST(1,I)) I1 = I                               MATP2590
400   IF ((I1.GT.0).AND.(I1.LT.6)) GO TO 410                       MATP2600
      IERROR(J,13)=1                                               MATP2610
      IERR=IERR+1                                                  MATP2620
      GO TO 430                                                    MATP2630
C                                                                  MATP2640
C     PERFORM INTERPOLATION                                        MATP2650
410   ALP=(ST(1,I1+1)-FA)/(ST(1,I1+1)-ST(1,I1))                    MATP2660
      BET=1.E0-ALP                                                 MATP2670
      DO 420 I=2,NST                                               MATP2680
```

```
      STS(I-1,J)=ALP*ST(I,I1)+BET*ST(I,I1+1)                            MATP2690
  420 STN(I-1,J)=ALP*ST(I+8,I1)+BET*ST(I+8,I1+1)                        MATP2700
  430 CONTINUE                                                          MATP2710
      EPSU(J)=STN(6,J)                                                  MATP2720
      GO TO 470                                                         MATP2730
C     *****************************************************************MATP2740
C     STORE USER PROVIDED STRESS-STRAIN CURVE                           MATP2750
C     *****************************************************************MATP2760
C                                                                       MATP2770
  440 DO 450 I=1,7                                                      MATP2780
  450 SLOPE(I,J)=99999.E0                                               MATP2790
      EPSU(J)=99999.E0                                                  MATP2800
      EC(J)=99999.E0                                                    MATP2810
      SLOPE(8,J)=99999.E0                                               MATP2820
      ET(J)=99999.E0                                                    MATP2830
      DO 460 I=2,7                                                      MATP2840
C                                                                       MATP2850
C     CHECK THAT ABSOLUTE VALUE OF STRAINS ARE IN ASCENDING ORDER       MATP2860
C                                                                       MATP2870
      IF (ABS(STN(I-1,J)).LE.ABS(STN(I,J))) GO TO 460                   MATP2880
      IERROR(J,15)=1                                                    MATP2890
      IERR=IERR+1                                                       MATP2900
      GO TO 560                                                         MATP2910
  460 CONTINUE                                                          MATP2920
C                                                                       MATP2930
C     DETERMINE VALUE OF STRAIN AT MAXIMUM STRESS                       MATP2940
      STSMAX=ABS(STS(1,J))                                              MATP2950
      NSTN=1                                                            MATP2960
      DO 465 K=2,7                                                      MATP2970
      IF (ABS(STS(K,J)).LE.STSMAX) GO TO 465                            MATP2980
```

```
      STSMAX=ABS(STS(K,J))
      NSTN=K
  465 CONTINUE
      EPSU(J)=STN(NSTN,J)
C
C     DETERMINE THE SLOPES OF THE SEGMENTS OF THE STRESS-STRAIN CURVES
C
470   DO 490 L=2,7
      TNUM=STS(L,J)-STS(L-1,J)
      TDEN=STN(L,J)-STN(L-1,J)
      IF (TDEN.NE.0.0) GO TO 480
      SLOPE(L-1,J)=1.0E0
      GO TO 490
480   SLOPE(L-1,J)=TNUM/TDEN
490   CONTINUE
      IF (ICODE(J).EQ.2) GO TO 510
      ET(J)=0.0E0
      EC(J)=SLOPE(1,J)
      GO TO 560
510   TNUM=STS(8,J)-STS(1,J)
      TDEN=STN(8,J)-STN(1,J)
      IF (TDEN.NE.0.0E0) GO TO 520
      SLOPE(7,J)=1.0E0
      GO TO 530
520   SLOPE(7,J)=TNUM/TDEN
530   TNUM=STS(3,J)-STS(8,J)
      TDEN=STN(3,J)-STN(8,J)
      IF (TDEN.NE.0.0E0) GO TO 540
      SLOPE(8,J)=1.0E0
      GO TO 550
```

```
540    SLOPE(8,J)=TNUM/TDEN                                              MATP3290
550    ET(J)=SLOPE(1,J)                                                  MATP3300
       EC(J)=SLOPE(1,J)                                                  MATP3310
560    CONTINUE                                                          MATP3320
       IF (G(J).NE.0.E0) GO TC 50                                        MATP3330
       IF (EC(J).NE.0.E0) G(J)=EC(J)/(2.E0*(1.E0+PR(J)))                 MATP3340
       IF (ET(J).NE.0.E0) G(J)=ET(J)/(2.E0*(1.E0+PR(J)))                 MATP3350
       GO TO 50                                                          MATP3360
570    NMAT=J                                                            MATP3370
       IIUNIT=IUNITS                                                     MATP3380
       NPAGE = 1                                                         MATP3390
580    CONTINUE                                                          MATP3400
C                                                                        MATP3410
C      IF NO DATA OUTPUT REQUIRED, SKIP DATA OUTPUT                      MATP3420
C                                                                        MATP3430
       IF (IPRINT.EQ.0) GO TO 610                                        MATP3440
C                                                                        MATP3450
C      WRITE GENERAL PROBLEM DESCRIPTION AND PAGE NUMBER                 MATP3460
       CALL PAGE                                                         MATP3470
C                                                                        MATP3480
C      WRITE DATA BLOCK HEADING                                          MATP3490
590    FORMAT (1H ,20A4,//)                                              MATP3500
       WRITE (NPRT,590) DHEAD                                            MATP3510
600    FORMAT (1H ,19X,19HMATERIAL CONSTANTS )                           MATP3520
       WRITE (NPRT,600)                                                  MATP3530
       LINE=LINE+2                                                       MATP3540
C                                                                        MATP3550
C      CHECK FOR NO INPUT TO DATA BLOCK                                  MATP3560
610    IF (NMAT.NE.0) GO TO 630                                          MATP3570
620    FORMAT (1H ,44H*** NO MATERIAL PROPERTIES INPUT (MATP). ***)      MATP3580
```

```
      PRINT 620                                                         MATP3590
      IERR=IERR+1                                                       MATP3600
      GO TO 1240                                                        MATP3610
  630 DO 1220 I=1,NMAT                                                  MATP3620
      IC=ICODE(I)+1                                                     MATP3630
C                                                                       MATP3640
C     IF NO DATA OUTPUT REQUIRED, SKIP DATA OUTPUT                      MATP3650
      IF (IPRINT.EQ.0) GO TO 640                                        MATP3660
      LINE=LINE+12                                                      MATP3670
      IF (LINE.LE.NL) GO TO 640                                         MATP3680
      CALL PAGE                                                         MATP3690
      WRITE (NPRT,600)                                                  MATP3700
      LINE=LINE+15                                                      MATP3710
C                                                                       MATP3720
C                                                                       MATP3730
C     CONVERT UNITS OF PRINT DATA.                                      MATP3740
  640 XEGSIS = 1.E0                                                     MATP3750
      XEGSIL = 1.E0                                                     MATP3760
      IF (NPAGE.EQ.2) GO TO 645                                         MATP3770
      XEGSIL = 1.E0/EGSIL                                               MATP3780
      XEGSIS = 1.E0/EGSIS                                               MATP3790
  645 DO 650 K=1,8                                                      MATP3800
      STSK(K) = STS(K,I)*XEGSIS                                         MATP3810
      IF (STS(K,I).EQ.99999.E0) STSK(K)=STS(K,I)                        MATP3820
  650 CONTINUE                                                          MATP3830
      FCFYI = FCFY(I)*XEGSIS                                            MATP3840
      IF (FCFY(I).EQ.99999.E0) FCFYI=FCFY(I)                            MATP3850
      GI = G(I)*XEGSIS                                                  MATP3860
      DENSI = DENS(I)*(XEGSIS/XEGSIL)                                   MATP3870
      ECI = EC(I)*XEGSIS                                                MATP3880
```

```
      ETI = ET(I)*XEGSIS                                                MATP3890
      IF (IPRINT.EQ.0) GO TO 890                                        MATP3900
      IF (IIUNIT.EQ.0.OR.IIUNIT.EQ.1) GO TO 780                         MATP3910
C                                                                       MATP3920
C     PRINT SI UNIT OUTPUT.                                             MATP3930
C                                                                       MATP3940
660   FORMAT (/,1H ,8HMATERIAL,3X,A4,17X,5A4)                           MATP3950
      WRITE (NPRT,660) NAME(I),(TYPA(J,IC),J=1,5)                       MATP3960
      IF (ICODE(I).EQ.2) GO TO 690                                      MATP3970
670   FORMAT (1H ,17HCRUSHING STRENGTH,13X,0PE15.7,2X,6HN/M**2,/,1H ,15HMATP3980
     1YOUNG*S MODULUS,15X,0PE15.7,2X,9HN/M.**2  )                       MATP3990
      WRITE(NPRT,670) FCFYI,ECI                                         MATP4000
      GO TO 710                                                         MATP4010
680   FORMAT (1H ,14HYIELD STRENGTH,16X,0PE15.7,2X,6HN/M**2,/,1H ,15HYOUMATP4020
     1NGS MODULUS ,15X,0PE15.7,2X,9HN/M.**2  )                          MATP4030
690   WRITE(NPRT,680) FCFYI,ETI                                         MATP4040
700   FORMAT (1H ,15HPOISSON*S RATIO,15X,0PE15.7,/,1H ,21HELASTIC SHEAR MATP4050
     1MODULUS,9X,0PE15.7,2X,6HN/M**2,/,1H ,16HMATERIAL DENSITY,14X,0PE15MATP4060
     2.7,2X,6HN/M**3,/,1H ,25HUNLOADING CURVE CONSTANTS,5X,0PE15.7,E15.7MATP4070
     3)                                                                 MATP4080
710   WRITE(NPRT,700) PR(I),GI,DENSI,STSK(8),STN(8,I)                   MATP4090
720   FORMAT (1H ,28HSTRESS-STRAIN CURVE POINTS- )                      MATP4100
      WRITE (NPRT,720)                                                  MATP4110
      IF (IC.NE.2) GO TO 760                                            MATP4120
730   FORMAT (1H ,20H  STRESS (N/M**2)    ,1X,4(0PE14.5),6X,A2,6X,2(0PE14MATP4130
     1.5))                                                              MATP4140
      WRITE (NPRT,730) (STSK(K),K=1,4),FT,STSK(6),STSK(7)               MATP4150
740   FORMAT (1H ,20H  STRAIN (M/M)       ,1X,4(0PE14.5),6X,A2,12X,A2,6X,MATP4160
     1(0PE14.5),/,76X,44H  $$THESE VALUES DEPEND ON STIRRUP SPACING. )  MATP4170
      WRITE (NPRT,740) (STN(K,I),K=1,4),FT,FT,STN(7,I)                  MATP4180
```

```
      GO TO 890                                                         MATP4190
750   FORMAT (1H ,20H  STRESS (N/M**2)    ,1X,7(0PE14.5))               MATP4200
760   WRITF (NPRT,750) (STSK(K),K=1,7)                                  MATP4210
770   FORMAT (1H ,20H  STRAIN  (M/M)      ,1X,7(0PE14.5))               MATP4220
      WRITE (NPRT,770) (STN(K,I),K=1,7)                                 MATP4230
      GO TO 890                                                         MATP4240
C                                                                       MATP4250
C     PRINT ENGLISH UNIT OUTPUT                                         MATP4260
C                                                                       MATP4270
780   WRITE (NPRT,660) NAME(I),(TYPA(J,IC),J=1,5)                       MATP4280
      IF (ICODE(I).EQ.2) GO TO 810                                      MATP4290
790   FORMAT (1H ,17HCRUSHING STRENGTH,13X,0PE15.7,2X,9HLB/IN.**2,/,1H ,MATP4300
     115HYOUNG*S MODULUS,15X,0PE15.7,2X,9HLB/IN.**2)                    MATP4310
      WRITE(NPRT,790) FCFYI,ECI                                         MATP4320
      GO TO 830                                                         MATP4330
800   FORMAT (1H ,14HYIELD STRENGTH,16X,0PE15.7,2X,9HLB/IN.**2,/,1H ,15HMATP4340
     1YOUNGS MODULUS ,15X,0PE15.7,2X,9HLB/IN.** 2)                      MATP4350
810   WRITE(NPRT,800) FCFYI,ETI                                         MATP4360
820   FORMAT (1H ,15HPOISSON*S RATIO,15X,0PE15.7,/,1H ,21HELASTIC SHEAR MATP4370
     1MODULUS,9X,0PE15.7,2X,9HLB/IN.**2,/,1H ,16HMATERIAL DENSITY,14X,1PMATP4380
     2E15.7,2X,9HLB/IN.**3,/,1H ,25HUNLOADING CURVE CONSTANTS,5X,0PE15.7MATP4390
     3,2X,0PE15.7)                                                      MATP4400
830   WRITE(NPRT,820) PR(I),GI,DENSI,STSK(8),STN(8,I)                   MATP4410
      WRITE (NPRT,720)                                                  MATP4420
      IF (IC.NE.2) GO TO 870                                            MATP4430
840   FORMAT (1H ,20H  STRESS (LB/IN.**2),1X,4(0PE14.5),6X,A2,6X,2(0PE14MATP4440
     1.5))                                                              MATP4450
      WRITE(NPRT,840) (STSK(K),K=1,4),FT,STSK(6),STSK(7)                MATP4460
850   FORMAT (1H ,20H  STRAIN (IN./IN.)  ,1X,4(0PE14.5),6X,A2,12X,A2,6X,MATP4470
     1(0PE14.5),/,76X,44H  $$THESE VALUES DEPEND ON STIRRUP SPACING. )  MATP4480
```

```
      WRITE (NPRT,850) (STN(K,I),K=1,4),FT,FT,STN(7,I)                  MATP4490
      GO TO 890                                                         MATP4500
860   FORMAT (1H ,20H  STRESS (LB/IN.**2),1X,7(0PE14.5))                MATP4510
870   WRITE(NPRT,860) (STSK(K),K=1,7)                                   MATP4520
880   FORMAT (1H ,20H  STRAIN (IN./IN.)  ,1X,7(0PE14.5))                MATP4530
      WRITE (NPRT,880) (STN(K,I),K=1,7)                                 MATP4540
C                                                                       MATP4550
C     OUTPUT ERROR MESSAGES                                             MATP4560
C                                                                       MATP4570
890   IF (IERROR(I,2).EQ.0) GO TO 940                                   MATP4580
900   FORMAT (1H ,52H*** UNRECOGNIZABLE MATERIAL CODE INPUT FOR MATERIALMATP4590
     1 ,A4,12H (MATP). ***)                                             MATP4600
      PRINT 900, NAME(I)                                                MATP4610
      GO TO 920                                                         MATP4620
910   LINE=LINE+1                                                       MATP4630
920   LINE=LINE+1                                                       MATP4640
      IF (LINE.LE.NL) GO TO 930                                         MATP4650
      CALL PAGE                                                         MATP4660
      IF (IPRINT.EQ.0) GO TO 930                                        MATP4670
      WRITE (NPRT,600)                                                  MATP4680
      LINE=LINE+2                                                       MATP4690
930   GO TO (940,960,980,990,1010,1030,1050,1070,                       MATP4700
     1       1090,1130,1150,1180,1200,1220), ITAG                       MATP4710
940   IF (IERROR(I,3).EQ.0) GO TO 960                                   MATP4720
950   FORMAT (1H ,63H*** UNRECOGNIZABLE STRESS-STRAIN CURVE TYPE INPUT FMATP4730
     1OR MATERIAL ,A4,12H (MATP). ***)                                  MATP4740
      PRINT 950, NAME(I)                                                MATP4750
      ITAG=2                                                            MATP4760
      GO TO 920                                                         MATP4770
 960  IF (IERROR(I,5).EQ.0) GO TO 990                                   MATP4780
```

```
 980  CONTINUE                                                            MATP4790
990   IF (IERROR(I,6).EQ.0) GO TO 1010                                    MATP4800
1000  FORMAT (1H ,52H*** NEGATIVE CONCRETE OFFSET SPECIFIED FOR MATERIALMATP4810
     1 ,A4,12H (MATP). ***)                                               MATP4820
      PRINT 1000, NAME(I)                                                 MATP4830
      ITAG=5                                                              MATP4840
      GO TO 920                                                           MATP4850
1010  IF (IERROR(I,7).EQ.0) GO TO 1030                                    MATP4860
1020  FORMAT (1H0,47H** VALUE OF POISSON+S RATIO INPUT FOR MATERIAL ,A4,MATP4870
     1        53H IS OUTSIDE ALLOWABLE LIMITS OF 0.0 TO 0.5 (MATP). **) MATP4880
      PRINT 1020, NAME(I)                                                 MATP4890
      ITAG=6                                                              MATP4900
      GO TO 920                                                           MATP4910
1030  IF (IERROR(I,8).EQ.0) GO TO 1050                                    MATP4920
1040  FORMAT (1H ,52H*** VALUE OF THE ELASTIC SHEAR MODULUS FOR MATERIALMATP4930
     1 ,A4,40H IS OUTSIDE ALLOWABLE LIMITS (MATP). ***)                   MATP4940
      PRINT 1040, NAME(I)                                                 MATP4950
      ITAG=7                                                              MATP4960
      GO TO 920                                                           MATP4970
1050  IF (IERROR(I,9).EQ.0) GO TO 1070                                    MATP4980
1060  FORMAT (1H0,30H** DENSITY INPUT FOR MATERIAL ,A4,64H LIES OUTSIDE MATP4990
     1ALLOWABLE LIMITS OF 0.0 TO 0.5 LB/IN**3 (MATP). **)                 MATP5000
      PRINT 1060, NAME(I)                                                 MATP5010
      ITAG=8                                                              MATP5020
      GO TO 920                                                           MATP5030
1070  IF (IERROR(I,10).EQ.0) GO TO 1090                                   MATP5040
1080  FORMAT (1H ,48H*** UNLOADING CURVE CONSTANT INPUT FOR MATERIAL ,A4MATP5050
     1,46H DOES NOT LIE BETWEEN 0.0 AND 1.0 (MATP).***      )             MATP5060
      PRINT 1080, NAME(I)                                                 MATP5070
      ITAG=9                                                              MATP5080
```

```
      GO TO 920                                                         MATP5090
 1090 IF (IERROR(I,11).EQ.0) GO TO 1130                                 MATP5100
 1100 FORMAT (1H ,59H*** VALUE OF CONCRETE CRUSHING STRENGTH INPUT FOR MMATP5110
     1ATERIAL ,A4,53H IS OUTSIDE ALLOWABLE LIMITS FOR INTERNALLY GENERATMATP5120
     2ED,/,1X,32HSTRESS-STRAIN CURVES (MATP). ***)                      MATP5130
      PRINT 1100, NAME(I)                                               MATP5140
      ITAG=10                                                           MATP5150
      GO TO 910                                                         MATP5160
 1130 IF (IERROR(I,13).EQ.0) GO TO 1150                                 MATP5170
 1140 FORMAT (1H ,37H***YIELD STRENGTH INPJT FOR MATERIAL ,A4,64H IS NOTMATP5180
     1 IN THE ACCEPTABLE RANGE OF 33000 TO 75000 PSI.(MATP)*** )        MATP5190
      PRINT 1140, NAME(I)                                               MATP5200
      ITAG = 11                                                         MATP5210
      GO TO 910                                                         MATP5220
 1150 IF (IERROR(I,14).EQ.0) GO TO 1180                                 MATP5230
 1160 FORMAT (1H ,46H*** NO STRESS-STRAIN CURVE INPUT FOR MATERIAL ,A4,1MATP5240
     12H (MATP). ***)                                                   MATP5250
      PRINT 1160, NAME(I)                                               MATP5260
      ITAG = 12                                                         MATP5270
      GO TO 920                                                         MATP5280
 1180 IF (IERROR(I,15).EQ.0) GO TO 1200                                 MATP5290
 1190 FORMAT (1H ,58H*** THE ABSOLUTE VALUES OF THE STRAINS INPUT FOR MAMATP5300
     1TERIAL ,A4,39H ARE NOT IN ASCENDING ORDER (MATP). ***)            MATP5310
      PRINT 1190, NAME(I)                                               MATP5320
      ITAG = 13                                                         MATP5330
      GO TO 920                                                         MATP5340
 1200 IF (IERROR(I,1).EQ.0) GO TO 1220                                  MATP5350
 1210 FORMAT (1H ,56H*** MORE MATERIALS INPUT THAN PROGRAM IS CODED TO HMATP5360
     1ANDLE,I3,12H (MATP). ***)                                         MATP5370
      PRINT 1210                                                        MATP5380
```

```
      ITAG = 14                                                         MATP5390
      GO TO 920                                                         MATP5400
 1220 CONTINUE                                                          MATP5410
C                                                                       MATP5420
C     CHECK FOR END OF OUTPUT                                           MATP5430
C                                                                       MATP5440
      IF (IIUNIT.EQ.0.OR.IIUNIT.EQ.2) GO TO 1240                        MATP5450
      IF (IERR.GT.IIERR) GO TO 1240                                     MATP5460
      IF (IPRINT.EQ.0) GO TO 1240                                       MATP5470
C                                                                       MATP5480
C     CHECK FOR MULTIPLE OUTPUT                                         MATP5490
C                                                                       MATP5500
      NPAGE = 2                                                         MATP5510
      IF (IIUNIT.EQ.1) IIUNIT = 2                                       MATP5520
      IF (IIUNIT.EQ.3) IIUNIT = 0                                       MATP5530
      GO TO 580                                                         MATP5540
C                                                                       MATP5550
C     CHANGE STORED STRESS-STRAIN CURVES TO ORIGINAL UNITS.             MATP5560
 1240 IF(IUNITS.EQ.0.OR.IUNITS.EQ.3) GO TO 2021                         MATP5570
      DO 2020 I=1,6                                                     MATP5580
      DO 2020 K=1,9                                                     MATP5590
 2020 ST(K,I) = ST(K,I)/EGSISS                                          MATP5600
 2021 RETURN                                                            MATP5610
      END                                                               MATP5620
```

```
CMATY    0  10
      SUBROUTINE MATY (INAME,MATN)
C
C     SUBROUTINE TO CALCULATE ELEMENT MATERIAL NUMBERS.
C
      COMMON /FIBER/ FDUM(399),ICODE(9),NAME(9)
      COMMON /MAINBK/ IDUM1(38),NMAT,IDUM2(10)
      MATN = 0
      DO 10 J=1,NMAT
      IF (INAME.EC.NAME(J)) MATN = J
10    CONTINUE
      RETURN
      END
```

```
CMEMB    0  10
      SUBROUTINE MEMB (M,UR,UD,IFLAG)                                   MEMB   0
C                                                                       MEMB  10
C     THIS SUBROUTINE GENERATES STRESS AND ENERGY CALCULATIONS FOR A    MEMB  20
C     GENERAL REINFORCED CONCRETE MEMBER(M).  THE CALCULATIONS ARE      MEMB  30
C     CONTROLLED BY (IFLAG), WHERE                                      MEMB  40
C     IFLAG=1, INDICATES THAT THE RECOVERABLE STRAIN ENERGY(UR)         MEMB  50
C              AND THE DISSIPATIVE STRAIN ENERGY (UD) ARE REQUIRED.     MEMB  60
C     IFLAG=3, INDICATES THAT STRESS RESULTS ARE REQUIRED FOR PRINT-    MEMB  70
C              OUT AND FOR UPDATING THE STRESS HISTORY DATA BANK.       MEMB  80
C     THE ELASTIC-PLASTIC STATUS OF THE MEMBER IS INDICATED BY          MEMB  90
C     MSTAT(M), WHERE                                                   MEMB 100
C                                                                       MEMB 110
C     MSTAT=1, INDICATES A MEMBER THAT IS RESTRICTED TO LINEAR          MEMB 120
C              ELASTIC RESPONSE.                                        MEMB 130
C     MSTAT=2, INDICATES A MEMBER THAT IS CURRENTLY LINEAR ELASTIC      MEMB 140
C              BUT MAY GO INELASTIC.                                    MEMB 150
C     MSTAT=3, INDICATES A MEMBER THAT IS INELASTIC.                    MEMB 160
C                                                                       MEMB 170
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),       MEMB 180
     1   MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),MEMB 190
     2   MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)                       MEMB 200
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),MEMB 210
     1   BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),     MEMB 220
     2   HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),MEMB 230
     3   TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                 MEMB 240
     4   XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45),  MEMB 250
     5   YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)               MEMB 260
      COMMON/STRNBK/SRP(4),SRQ(4),UX,UY,UZ,XLEN,AREA,ZZI,IMAT           MEMB 270
C                                                                       MEMB 280
```

```
      XLEN=XL(M)
      IMAT=MATR(M)
      UR = 0.E0
      UD = 0.E0
C
C     CHECK FOR INELASTIC ELEMENT.
      IF (MSTAT(M).EQ.3) UD = UDM(M)
C
C     DETERMINE LOCAL DEFORMATIONS.
      CALL DEFO (M)
C
C     DETERMINE CONCRETE ENERGY CONTRIBUTION.
C
      CALL COEN (M,UR,UD,IFLAG)
C
C     DETERMINE LONGITUDINAL STEEL ENERGY CONTRIBUTION.
C
      CALL STEN (M,UR,UD,IFLAG)
C
C     STORE TOTAL MEMBER ENERGY.
C
      IF (IFLAG.NE.3) GO TO 30
      URM(M)=UR
      UDM(M)=UD
30    RETURN
      END
```

```
COUTS    0  10
      SUBROUTINE OUTS                                                   OUTS   0
C                                                                       OUTS  10
C     THIS SUBROUTINE REPORTS ANALYSIS RESULTS AT THE END OF EACH TIME  OUTS  20
C     STEP AS FOLLOWS:                                                  OUTS  30
C     JOINT STATUS: DISPLACEMENT, VELOCITY AND ACCELERATION; REPORTED   OUTS  40
C     .     FOR ALL OUTPUT OPTIONS.                                     OUTS  50
C     INTERNAL ENERGY DISTRIBUTION, BY ELEMENTS; REPORTED FOR ALL       OUTS  60
C     .     OUTPUT OPTIONS.                                             OUTS  70
C     AVERAGE STRESSES AND STRAINS, BY ELEMENTS; REPORTED ONLY FOR      OUTS  80
C     .     OUTPUT OPTIONS ISTRES = +S+ OR +B+.                         OUTS  90
C     ELEMENT END FORCES, BY ELEMENTS; REPORTED ONLY FOR OUTPUT         OUTS 100
C     .     OPTIONS ISTRES = +R+ OR +B+.                                OUTS 110
C     SOLUTION ACCURACY - OVERALL ENERGY DISTRIBUTION FOR ALL OUTPUT    OUTS 120
C     .     OPTIONS; ALSO REPORTED BY JOINTS FOR OUTPUT OPTIONS         OUTS 130
C     .     IPRINT = +D+ OR +E+.                                        OUTS 140
C                                                                       OUTS 150
      COMMON DATA(10000),KDATA(500)                                     OUTS 160
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), OUTS 170
     1     ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),     OUTS 180
     2     XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)    OUTS 190
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), OUTS 200
     1     PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB     OUTS 210
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ (45),IQL(20),MATR(45),      OUTS 220
     1     MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),OUTS 230
     2     MTYPE(45 ),NGRP(45),NSPAC(6,45),NTIES(45)                    OUTS 240
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9), OUTS 250
     1     SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)OUTS 260
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,OUTS 270
     1     IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,OUTS 280
```

```
     2    NCRD,NDF,NDFO,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,  OUTS 290
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,   OUTS 300
     4    NTIMES,NVEL,IINITD                                             OUTS 310
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),OUTS 320
     1   BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),      OUTS 330
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),OUTS 340
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                 OUTS 350
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45),  OUTS 360
     5    YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)               OUTS 370
      COMMON/PRNTBK/MPRINT                                               OUTS 380
      COMMON/SAVEBK/SAVACC(3,50),SAVAXL(2,45),SAVCRV(2,45),SAVMOM(2,45)  OUTS 390
     1    ,SAVSHR(2,45),SAVSRP(3,20),SAVSRQ(3,20),SAVXDJ(3,50),          OUTS 400
     2    SAVVEL(3,50),SVSTRN(12,45),SVSTRS(12,45)                       OUTS 410
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,           OUTS 420
     1     LTABI,NMAX,NMAXI                                              OUTS 430
      COMMON/SEEKBK/DEFOR(90),STPSIZ(90),GRAD(90),GRADI(90),DELTAG(90),  OUTS 440
     1     DIRECT(90),DIAG(90),STEP( 4),DSTEP(4),FVAL(4),VALUES(7),      OUTS 450
     2     DISACC,SSIZE,FUNACC,FUNMIN,CRITL,CRITU,NLIN                   OUTS 460
      COMMON/STRNBK/SRP(4),SRQ(4),UX,UY,UZ,XLEN,AREA,ZZI,IMAT            OUTS 470
      DIMENSION TEMP(24),STRESS(30),STRAIN(30),PCTYLD(24),PCTFX(6)       OUTS 480
     1          ,ENERGY(5,4),ETYPE(3,5),LOCA(3, 3)                       OUTS 490
      INTEGER EUNIT(2,2),FUNIT(4,2),SUNIT(3,2),AD(3 ),LENGTH(2),         OUTS 500
     2     LABELS(6),KOUT(6),LSA(6),LSB(6),LABEL(5 ,6),ELA,PLA,TIC,CRA,  OUTS 510
     3     KED,RUP,URD,CRU,HED,HEAD,DHEAD                                OUTS 520
      DATA LENGTH/3H M.,3HIN./,LABELS/4HUPPE,4HR FL,4HANGE,4HLOWE,       OUTS 530
     2     4HR FL,4HANGE/,FUNIT/4HNEWT,4HONS,,4H MET,4HERS ,4HPOUN,      OUTS 540
     3     4HDS, ,4HINCH,4HES  /,EUNIT/4HNT.-,4H M. ,4HIN.-,4H LB./,     OUTS 550
     4     SUNIT/4H NT.,4H/M.*,4H*2  ,4H LB.,4H/IN.,4H**2 /,AD/2HDX,     OUTS 560
     5     2HDY,2HRZ/,LABEL/4HSTEE,4HL, U,4HPPER,4H    ,4H    ,4HSTEE,   OUTS 570
     6     4HL, L,4HOWER,4H    ,4H    ,4HEDGE,4H CON,4HC., ,4HUPPE,      OUTS 580
```

```
     7     4HR   ,4HCONF,4H. CO,4HNC.,,4H UPP,4HER  ,4HCONF,4H. CO,     OUTS 590
     8     4HNC.,,4H LOW,4HER  ,4HEDGE,4H CON,4HC., ,4HLOWE,4HR   /,    OUTS 600
     9     ELA,PLA,TIC,CRA,KED,RUP,URD,CRU,HED/4HELAS,4HPLAS,4HTIC ,      OUTS 610
     *     4HCRAC,4HKED ,4HRUPT,4HURED,4HCRUS,4HHED /                     OUTS 620
      DATA ETYPE/4HRECO,4HVERA,4HBLE ,4HDISS,4HIPAT,4HIVE ,4HKINE,        OUTS 630
     1          4HTIC ,4H    ,4HTOTA,4HL   ,4H    ,4HEXTE,4HRNAL,4H    / OUTS 640
      DATA LOCA/4HUNCO,4HNF. ,4HCONC,4HCONF,4H. CO,4HNC. ,4HREIN,         OUTS 650
     1          4HF. S,4HTEEL/                                            OUTS 660
C                                                                         OUTS 670
      NDIM = 2                                                            OUTS 680
      IF (IUNITS.EQ.1.OR.IUNITS.EQ.2) NDIM = 1                            OUTS 690
C                                                                         OUTS 700
C     *******************************************************************OUTS 710
C     ************* THIS SEGMENT REPORTS THE JOINT STATUS **************OUTS 720
C     *******************************************************************OUTS 730
C     AD(3) = LABELS FOR DIRECTION COMPONENTS                             OUTS 740
C     XDJ(I,J)*DMNSLZ = DISPLACEMENT AT JOINT J IN THE I-DIRECTION        OUTS 750
C     VEL(I,J)*DMNSLZ = VELOCITY AT JOINT J IN THE I-DIRECTION            OUTS 760
C     ACC(I,J)*DMNSLZ = ACCELERATION AT JOINT J IN THE I-DIRECTION        OUTS 770
C                                                                         OUTS 780
      LINE = NL                                                           OUTS 790
      DO 200  JTNUM=1,NJ                                                  OUTS 800
      LINE = LINE + 4                                                     OUTS 810
      IF(LINE.LT.NL)  GO TO 90                                            OUTS 820
      CALL PAGE                                                           OUTS 830
      LINE = LINE + 10                                                    OUTS 840
      WRITE (NPRT,6000) TIME,LENGTH (NDIM),LENGTH(NDIM),LENGTH (NDIM)     OUTS 850
 6000 FORMAT(29X,15HJOINT STATUS AT,0PE11.4,8H SECONDS//7X,9HJOINT AND,  OUTS 860
     2     10X,12HDISPLACEMENT,12X,8HVELOCITY,11X,12HACCELERATION/7X,     OUTS 870
     3     9HDIRECTION,11X,A3,6H(RAD.),11X,A3,11H(RAD.)/SEC.,5X,A3,       OUTS 880
```

```
    4      16H(RAD.)/SEC./SEC./7X,9H---------,10X,12H------------,4X,  OUTS 890
    5      2(5X,14H--------------),5H-----/)                           OUTS 900
   90 DO 100 I=1,3                                                     OUTS 910
C                                                                      OUTS 920
C     CONVERT TO STRUCTURE SCALE FOR OUTPUT                            OUTS 930
      DMNSLZ = AVGL                                                    OUTS 940
      IF(I.EQ.3) DMNSLZ = 1.E0                                         OUTS 950
      TEMP(1) = XDJ(I,JTNUM)*DMNSLZ                                    OUTS 960
      SAVXDJ(I,JTNUM)=TEMP(1)                                          OUTS 970
      TEMP(2) = VEL(I,JTNUM)*DMNSLZ                                    OUTS 980
      IF(ABS(VEL(I,JTNUM)-TINY).LE.EPS) TEMP(2)=0.E0                   OUTS 990
      SAVVEL(I,JTNUM)=TEMP(2)                                          OUTS1000
      TEMP(3) = ACC(I,JTNUM)*DMNSLZ                                    OUTS1010
      IF(ABS(ACC(I,JTNUM)-TINY).LE.EPS) TEMP(3)=0.E0                   OUTS1020
      SAVACC(I,JTNUM)=TEMP(3)                                          OUTS1030
      WRITE(NPRT,6010) JTNUM,AD(I),(TEMP(K),K=1,3)                     OUTS1040
 6010 FORMAT(8X,I2,3X,A2,3(11X,1PE11.4))                               OUTS1050
  100 CONTINUE                                                         OUTS1060
      WRITE(NPRT,6020)                                                 OUTS1070
 6020 FORMAT(1H )                                                      OUTS1080
  200 CONTINUE                                                         OUTS1090
C                                                                      OUTS1100
C     PRINT OUT LOCAL ELEMENT DISTORTIONS FOR EXTENSIVE OUTPUT.         OUTS1110
C                                                                      OUTS1120
      IF(IPRINT.LT.3) GO TO 399                                        OUTS1130
      LINE = NL                                                        OUTS1140
      DO 300 M=1,NM                                                    OUTS1150
      XLEN = XL(M)                                                     OUTS1160
      LINE = LINE + 1                                                  OUTS1170
      IF (LINE.LT.NL) GO TO 210                                        OUTS1180
```

```
      CALL PAGE                                                         OUTS1190
      LINE = LINE + 4                                                   OUTS1200
      WRITE(NPRT,6025) LENGTH(NDIM),TIME                                OUTS1210
 6025 FORMAT(15X,27HLOCAL ELEMENT DISTORTIONS (,A3,10H,RAD.) AT ,1PE14.7OUTS1220
     1,8H SECONDS//5X,7HELEMENT,6X,5HNODES,6X,9HXDM OR U4,9X,8HUX OR U1,OUTS1230
     29X,8HUY OR U2,9X,8HUZ OR U3/)                                     OUTS1240
  210 CALL DEFO(M)                                                      OUTS1250
  300 WRITE(NPRT,6026) M,IP(M),IQ(M),XDM(M),UX,UY,UZ                    OUTS1260
 6026 FORMAT(8X,I2,6X,I2,4H TO ,I2,4(3X,1PE14.7))                       OUTS1270
C                                                                       OUTS1280
C     ******************************************************************OUTS1290
C     ***** THIS SEGMENT REPORTS THE INTERNAL ENERGY DISTRIBUTION ******OUTS1300
C     ******************************************************************OUTS1310
C           IP(M) = JOINT AT P-END OF MEMBER M                          OUTS1320
C           IQ(M) = JOINT AT Q-END OF MEMBER M                          OUTS1330
C          URM(M) = RECOVERABLE ENERGY OF MEMBER M                      OUTS1340
C          UDM(M) = DISSIPATED ENERGY OF MEMBER M                       OUTS1350
C           SYSUR = TOTAL RECOVERABLE SYSTEM ENERGY                     OUTS1360
C           SYSUD = TOTAL DISSIPATED SYSTEM ENERGY                      OUTS1370
C           SYSU  = TOTAL SYSTEM ENERGY                                 OUTS1380
C                                                                       OUTS1390
  399 SYSUR = 0.E0                                                      OUTS1400
      SYSUD = 0.E0                                                      OUTS1410
      SYSU  = 0.E0                                                      OUTS1420
      LINE = NL                                                         OUTS1430
      DO 400 M=1,NM                                                     OUTS1440
      IF(MTYPE(M).NE.4) CALL MEMB(M,UR,UD,1)                            OUTS1450
      IF(MTYPE(M).EQ.4) CALL WIDE(M,UR,UD,1)                            OUTS1460
      SYSUR = SYSUR + URM(M)                                            OUTS1470
      SYSUD = SYSUD + UDM(M)                                            OUTS1480
```

```
  400 CONTINUE                                                          OUTS1490
      IF(NLS.EQ.0) GO TO 420                                            OUTS1500
      DO 410 L=1,NLS                                                    OUTS1510
      CALL LEAF(-L,UR,1)                                                OUTS1520
      SYSUR = SYSUR + UR                                                OUTS1530
  410 CONTINUE                                                          OUTS1540
  420 SYSU = SYSUR+SYSUD                                                OUTS1550
      IF(SYSUR.EQ.0.E0) SYSUR=TINY                                      OUTS1560
      IF(SYSUD.EQ.0.E0) SYSUD=TINY                                      OUTS1570
      IF(SYSU.EQ.0.E0) SYSU=TINY                                        OUTS1580
      DO 500 M=1,NM                                                     OUTS1590
      TEMP(1) = URM(M)                                                  OUTS1600
      TEMP(2) = UDM(M)                                                  OUTS1610
      TEMP(3) = TEMP(1)+TEMP(2)                                         OUTS1620
      TEMP(4) = TEMP(1)/SYSUR                                           OUTS1630
      TEMP(5) = TEMP(2)/SYSUD                                           OUTS1640
      TEMP(6) = TEMP(3)/SYSU                                            OUTS1650
      LINE = LINE + 2                                                   OUTS1660
      IF(LINE.LT.NL) GO TO 490                                          OUTS1670
      CALL PAGE                                                         OUTS1680
      LINE = LINE + 6                                                   OUTS1690
      WRITE(NPRT,6030) (EUNIT(J,NDIM),J=1,2),TIME                       OUTS1700
 6030 FORMAT(18X,17HINTERNAL ENERGY (,2A4,21H) - - DISTRIBUTION AT,     OUTS1710
     *    1PE11.4,8H SECONDS//                                          OUTS1720
     2    1X,7HELEMENT,4X,11HRECOVERABLE,5X,10HDISSIPATED,7X,7HELEMENT,OUTS1730
     3    6X,3(10HPERCENT OF,5X)/1X,8HIDENTITY,6X,2(6HENERGY,9X),       OUTS1740
     4    6HENERGY,7X,11HRECOVERABLE,4X,10HDISSIPATED,4X,10HPARTICIPAT,OUTS1750
     5    3HION/1X,8H--------,6(3X,12H------------),1H-/)               OUTS1760
  490 WRITE(NPRT,6040) IP(M),IQ(M),(TEMP(N),N=1,6)                      OUTS1770
 6040 FORMAT(2X,I2,1H-,I2,1X,3(4X,0PE11.4),6X,3(2PF8.3,7X)/)            OUTS1780
```

```
  500 CONTINUE                                                          OUTS1790
C                                                                       OUTS1800
      IF(ISTRES.NE.1.AND.ISTRES.NE.3 ) GO TO 2999                       OUTS1810
C                                                                       OUTS1820
C     ******************************************************************OUTS1830
C     *********** THIS SEGMENT REPORTS AVERAGE STRESSES AND ************OUTS1840
C     *********** STRAINS FOR A REINFORCED CONCRETE ELEMENT ************OUTS1850
C     ******************************************************************OUTS1860
C                                                                       OUTS1870
      LINE = NL                                                         OUTS1880
      DO 1110 M=1,NM                                                    OUTS1890
C                                                                       OUTS1900
C     VERIFY ELEMENT TYPE                                               OUTS1910
      IF(MTYPE(M).GE.4) GO TO 1110                                      OUTS1920
      NGRPM = NGRP(M)                                                   OUTS1930
      XLEN = XL(M)                                                      OUTS1940
      AREA = HMEM(M)*BMEM(M)                                            OUTS1950
C                                                                       OUTS1960
C     FIND GROUP NUMBERS OF UPPER AND LOWER STEEL GROUPS                OUTS1970
      TEMPA = HTOP(M)-D(M)                                              OUTS1980
      TEMPB = HTOP(M)-DP(M)                                             OUTS1990
      ITOP = 1                                                          OUTS2000
      IBOT = 1                                                          OUTS2010
      DO 600 I=1,NGRPM                                                  OUTS2020
      IF(ABS((YBAR(I,M))-TEMPA).LE.1.E-2) IBOT = I                      OUTS2030
      IF(ABS((YBAR(I,M))-TEMPB).LE.1.E-2) ITOP = I                      OUTS2040
  600 CONTINUE                                                          OUTS2050
C                                                                       OUTS2060
C     COLLECT MEMBER DIMENSIONS AND MATERIAL PROPERTIES                 OUTS2070
      TEMP(1) = YBAR(ITOP,M)                                            OUTS2080
```

```
      TEMP(2)  = YBAR(IBOT,M)                                           OUTS2090
      TEMP(3)  = YFIBR( 1,M)                                            OUTS2100
      TEMP(4)  = YFIBR( 2,M)                                            OUTS2110
      TEMP(5)  = YFIBR(10,M)                                            OUTS2120
      TEMP(6)  = YFIBR(11,M)                                            OUTS2130
      KOUT(1)  = MBAR(ITOP,M)                                           OUTS2140
      KOUT(2)  = MBAR(IBOT,M)                                           OUTS2150
      KOUT(3)  = MCODE (M)                                              OUTS2160
      KOUT(4)  = MATR  (M)                                              OUTS2170
      KOUT(5)  = MATR  (M)                                              OUTS2180
      KOUT(6)  = MCODE (M)                                              OUTS2190
      DO 1100 L=1,2                                                     OUTS2200
      IF(MSTAT(M).EQ.3) GO TO 900                                       OUTS2210
C                                                                       OUTS2220
C     STRESS AND STRAIN COMPUTATIONS FOR AN ELASTIC MEMBER              OUTS2230
      XLOC = 0.E0                                                       OUTS2240
      IF(L.EQ.2) XLOC = XLEN                                            OUTS2250
      CALL DEFO(M)                                                      OUTS2260
      DO 800 K=1,6                                                      OUTS2270
      CALL STRN(M,XLOC,TEMP(K),STRAIN(K))                               OUTS2280
      LSA(K) = ELA                                                      OUTS2290
      LSB(K) = TIC                                                      OUTS2300
      IF(K.GE.3.AND.STRAIN(K).GT.0.E0) GO TO 720                        OUTS2310
      J = KOUT(K)                                                       OUTS2320
      STRESS(K) = EC(J)*STRAIN(K)                                       OUTS2330
      IF (STRAIN(K).GT.0.E0) STRESS(K) = ET(J)*STRAIN(K)                OUTS2340
      PCTYLD(K) = ABS(STRESS(K)/FCFY(J))                                OUTS2350
      IF(PCTYLD(K).GE.1.E0) PCTYLD(K) = 1.E0                            OUTS2360
      PCTFX(K)  = ABS(STRAIN(K)/STN(7,J))                               OUTS2370
      IF(PCTFX(K).GE.1.E0) PCTFX(K) = 1.E0                              OUTS2380
```

```
      GO TO 800                                                         OUTS2390
  720 STRESS(K) = 0.E0                                                  OUTS2400
      PCTYLD(K) = 1.E0                                                  OUTS2410
      PCTFX(K)  = 1.E0                                                  OUTS2420
      LSA(K) = CRA                                                      OUTS2430
      LSB(K) = KED                                                      OUTS2440
  800 CONTINUE                                                          OUTS2450
      GO TO 1070                                                        OUTS2460
C                                                                       OUTS2470
C     STRESS AND STRAIN DATA COLLECTION FOR YIELDED MEMBERS             OUTS2480
C                                                                       OUTS2490
 900  INDEX = KDATA(LPI+M)-1                                            OUTS2500
      STRAIN(1) = DATA(INDEX+NGRPM*3+(ITOP-1)*2+L)                      OUTS2510
      STRAIN(2) = DATA(INDEX+NGRPM*3+(IBOT-1)*2+L)                      OUTS2520
      STRAIN(3) = DATA(INDEX+NGRPM*5+(L-1)*5+22)                        OUTS2530
      STRAIN(4) = DATA(INDEX+NGRPM*5+(L-1)*9+41)                        OUTS2540
      STRAIN(5) = DATA(INDEX+NGRPM*5+(L-1)*9+49)                        OUTS2550
      STRAIN(6) = DATA(INDEX+NGRPM*5+(L-1)*5+26)                        OUTS2560
C                                                                       OUTS2570
C     STRESSES AND FAILURE CRITERIA IN REINFORCING STEEL                OUTS2580
      DO 980 K=1,2                                                      OUTS2590
      J = KOUT(K)                                                       OUTS2600
C                                                                       OUTS2610
C     COMPUTE SUBSCRIPTS                                                OUTS2620
      IBOTOP = ITOP                                                     OUTS2630
      IF(K.EQ.2) IBOTOP = IBOT                                          OUTS2640
      INDEX = KDATA(LPSI+M) - 1 + NGRPM*24 + (IBOTOP-1)*16 + (L-1)*8    OUTS2650
C                                                                       OUTS2660
C     COLLECT STRESS HISTORY DATA                                       OUTS2670
      DO 940 KK=1,8                                                     OUTS2680
```

```
      S(KK) = DATA(INDEX+KK)                                            OUTS2690
  940 CONTINUE                                                          OUTS2700
      IF (S(1).LE.0.5E0) GO TO 960                                      OUTS2710
      PCTYLD(K) = 1.E0                                                  OUTS2720
      PCTFX(K)  = 1.E0                                                  OUTS2730
      STRESS(K) = 0.E0                                                  OUTS2740
      LSA(K) = RUP                                                      OUTS2750
      LSB(K) = URD                                                      OUTS2760
      GO TO 980                                                         OUTS2770
  960 POSNEG = -1.E0                                                    OUTS2780
      IF((STRAIN(K)-S(5)).GE.0.E0) POSNEG = 1.E0                        OUTS2790
      STRESS(K) = S(8)*POSNEG                                           OUTS2800
      LSB(K) = TIC                                                      OUTS2810
      LSA(K) = ELA                                                      OUTS2820
      IF(ABS(S(6)).GE.1.E-4) LSA(K) = PLA                               OUTS2830
      KM = INT(S(2))*6 + 2                                              OUTS2840
      PCTYLD(K) = S(8)/STS(KM,J)                                        OUTS2850
      IF(PCTYLD(K).GE.1.E0) PCTYLD(K) = 1.E0                            OUTS2860
      PCTFX(K) = (STRAIN(K)-S(5))/(POSNEG*STN(7,J))                     OUTS2870
      IF(PCTFX(K).GE.1.E0) PCTFX(K) = 1.E0                              OUTS2880
  980 CONTINUE                                                          OUTS2890
C                                                                       OUTS2900
C     STRESSES AND FAILURE CRITERIA IN CONCRETE                         OUTS2910
C                                                                       OUTS2920
      DO 1060 K=3,6                                                     OUTS2930
      J = KOUT(K)                                                       OUTS2940
C                                                                       OUTS2950
C     COMPUTE SUBSCRIPTS                                                OUTS2960
      KA = 0                                                            OUTS2970
      KB = 40                                                           OUTS2980
```

```
      KC = 32                                                           OUTS2990
      IF(K.EQ.3.OR.K.EQ.6) GO TO 990                                    OUTS3000
      KA = 152                                                          OUTS3010
      KB = 72                                                           OUTS3020
      KC = 64                                                           OUTS3030
  990 IF(K.EQ.3.OR.K.EQ.4) KC = 0                                       OUTS3040
      INDEX = KDATA(LPSI+M) -1+NGRP(M)*40+KA+KC+(L-1)*KB+168            OUTS3050
C                                                                       OUTS3060
C     COLLECT STRESS HISTORY DATA                                       OUTS3070
      DO 1000 KK=1,8                                                    OUTS3080
      S(KK) = DATA(INDEX+KK)                                            OUTS3090
 1000 CONTINUE                                                          OUTS3100
      STRESS(K) = S(6)                                                  OUTS3110
      IF(STRAIN(K).GT.(S(8)+STN(1,J))) GO TO 1040                       OUTS3120
      IF(STRAIN(K).LT.STN(7,J)) GO TO 1030                              OUTS3130
      LSB(K) = TIC                                                      OUTS3140
      IF(S(8).GT.0.E0) GO TO 1010                                       OUTS3150
      LSA(K) = ELA                                                      OUTS3160
      PCTYLD(K) = STRESS(K)/STS(2,J)                                    OUTS3170
      IF(PCTYLD(K).GE.1.E0) PCTYLD(K) = 1.E0                            OUTS3180
      GO TO 1020                                                        OUTS3190
 1010 LSA(K) = PLA                                                      OUTS3200
      PCTYLD(K) = 1.E0                                                  OUTS3210
 1020 PCTFX(K) = STRAIN(K)/STN(7,J)                                     OUTS3220
      IF(PCTFX(K).GE.1.E0) PCTFX(K) = 1.E0                              OUTS3230
      GO TO 1060                                                        OUTS3240
 1030 LSA(K) = CRU                                                      OUTS3250
      LSB(K) = HED                                                      OUTS3260
      GO TO 1050                                                        OUTS3270
 1040 LSA(K) = CRA                                                      OUTS3280
```

```
      LSB(K) = KED                                                      OUTS3290
 1050 PCTYLD(K) = 1.E0                                                  OUTS3300
      PCTFX(K)  = 1.E0                                                  OUTS3310
 1060 CONTINUE                                                          OUTS3320
C                                                                       OUTS3330
C     PRINTING SECTION                                                  OUTS3340
C                                                                       OUTS3350
1070   DO 1075 K=1,6                                                    OUTS3360
      SVSTRS((L-1)*6+K,M) = STRESS(K)                                   OUTS3370
      SVSTRN((L-1)*6+K,M) = STRAIN(K)                                   OUTS3380
1075   CONTINUE                                                         OUTS3390
      LINE = LINE + 7                                                   OUTS3400
      IF(LINE.LT.NL) GO TO 1080                                         OUTS3410
      CALL PAGE                                                         OUTS3420
      LINE = LINE + 16                                                  OUTS3430
      WRITE(NPRT,6050)  (SUNIT(J,NDIM),J=1,3), TIME                     OUTS3440
 6050 FORMAT(14X,18HAVERAGE STRESSES (,3A4,                             OUTS3450
     *     33H) IN REINFORCED CONCRETE ELEMENTS,                        OUTS3460
     2     3H AT,1PE11.4,8H SECONDS//21X,11HSIT E IN THE,2X,2(9X,6HNORMALOUTS3470
     3     ),6X,7HLOADING,2(4X,7HPERCENT),3H OF/1X,3HEND,3X,7HELEMENT,6XOUTS3480
     4     ,13HCROSS SECTION,10X,6HSTRESS,9X,6HSTRAIN,7X,5HSTATE,4X,    OUTS3490
     5     9HOF  YIELD,4X,8HFRACTURE/1X,3H---,3X,7H-------,3X,7H-------,OUTS3500
     6     13H-------------,2(3X,12H------------),3X,7H-------,3X,      OUTS3510
     7     9H---------,3X,10H----------/)                               OUTS3520
 1080 IPQ = IP(M)                                                       OUTS3530
      IF(L.EQ.2) IPQ = IQ(M)                                            OUTS3540
      WRITE(NPRT,6060 ) IPQ,IP(M),IQ(M), ((LABEL(J,K),J=1,5),STRESS(K), OUTS3550
     2     STRAIN(K),LSA(K),LSB(K ),PCTYLD(K),PCTFX(K),K=1,6)           OUTS3560
 6060 FORMAT(1X,I2,5X,I2,1H-,I2,6(T18,5A4,2X,2(1PE11.4,4X),2A4,2X,      OUTS3570
     2     2 (2PF8.3,5X))/ )                                            OUTS3580
```

```
 1100 CONTINUE
 1110 CONTINUE
      IF (IPRINT.LT.3) GO TO 1999
C
C     PRINT STRESSES AND STRAINS AT GAUSS POINTS FOR EXTENSIVE OUTPUT.
C
      LINE = NL
      DO 1800 M=1,NM
C
C     VERIFY ELEMENT TYPE.
      IF(MTYPE(M).GE.4) GO TO 1800
      NGRPM=NGRP(M)
      XLEN = XL(M)
      IF(MSTAT(M).NE.3) CALL CEFO(M)
      DO 1790 K=1,3
      IL =0
      GO TO (1710,1720,1730),K
C     UNCONFINED CONCRETE.
1710  KN = 7
      J = MCODE(M)
      IF (MSTAT(M).EQ.3) GO TO 1715
C     ELASTIC COMPUTATIONS.
1711  DO 1714 I=1,7
      IF(K.EQ.2.AND.(I.LE.2.OR.I.GE.6)) GO TO 1714
      DO 1712 L=1,3
      IL = IL +1
      CALL STRN(M,XPI(L,M),YGP(I,M),STRAIN(IL))
      ELASMD = ET(J)
      IF (STRAIN(IL).LT.0.E0) ELASMD = EC(J)
      STRESS(IL) = ELASMD*STRAIN(IL)
```

```
 1712 CONTINUE                                                          OUTS3890
 1714 CONTINUE                                                          OUTS3900
      GO TO 1740                                                        OUTS3910
 1715 KRESS = KDATA(LPSI+M) - 1 + 40*NGRP(M)                            OUTS3920
      KRAIN = KDATA(LPI+M) - 1 + 5*NGRP(M)                              OUTS3930
 1716 IL = 3*KN                                                         OUTS3940
      DO 1718 L=1,IL                                                    OUTS3950
      STRAIN(L) = DATA(KRAIN+L)                                         OUTS3960
      STRESS(L) = DATA(KRESS+8*L-2)                                     OUTS3970
      IF (STRAIN(L).GT.(STN(1,J)+DATA(KRESS+8*L))) STRESS(L) = 0.E0     OUTS3980
 1718 CONTINUE                                                          OUTS3990
      GO TO 1740                                                        OUTS4000
C     CONFINED CONCRETE.                                                OUTS4010
 1720 KN = 3                                                            OUTS4020
      J = MATR(M)                                                       OUTS4030
      IF (MSTAT(M).NE.3) GO TO 1711                                     OUTS4040
      KRESS = KDATA(LPSI+M) - 1 + 40*NGRP(M) + 248                      OUTS4050
      KRAIN = KDATA(LPI+M) - 1 + 5*NGRP(M) + 31                         OUTS4060
      GO TO 1716                                                        OUTS4070
C     REINFORCING STEEL.                                                OUTS4080
 1730 KN = NGRP(M)                                                      OUTS4090
      KRESS = KDATA(LPSI+M) - 1                                         OUTS4100
      KRAIN = KDATA(LPI+M) - 1                                          OUTS4110
      IF (KN.EQ.0) GO TO 1790                                           OUTS4120
      DO 1734 I=1,KN                                                    OUTS4130
      J = MBAR(I,M)                                                     OUTS4140
      DO 1733 L=1,3                                                     OUTS4150
      IL = IL + 1                                                       OUTS4160
      IF (MSTAT(M).EQ.3) GO TO 1732                                     OUTS4170
      CALL STRN(M,XPI(L,M),YBAR(I,M),STRAIN(IL))                        OUTS4180
```

```
      ELASMD = ET(J)                                                    OUTS4190
      IF (STRAIN(IL).LT.0.E0) ELASMD = EC(J)                            OUTS4200
      STRESS(IL) = ELASMD*STRAIN(IL)                                    OUTS4210
      GO TO 1733                                                        OUTS4220
 1732 STRAIN(IL) = DATA(KRAIN+IL)                                       OUTS4230
      STRESS(IL) = DATA(KRESS+IL*8)                                     OUTS4240
      IF (STRAIN(IL).LT.DATA(KRESS+IL*8-3)) STRESS(IL)=-STRESS(IL)      OUTS4250
 1733 CONTINUE                                                          OUTS4260
 1734 CONTINUE                                                          OUTS4270
C                                                                       OUTS4280
C     PRINT OUT HEADING.                                                OUTS4290
 1740 LINE = LINE + KN + 1                                              OUTS4300
      IF (LINE.LT.NL) GO TO 1750                                        OUTS4310
      CALL PAGE                                                         OUTS4320
      LINE = LINE + KN + 5                                              OUTS4330
      WRITE(NPRT,6070) (SUNIT(JS,NOIM),JS=1,3),TIME                     OUTS4340
 6070 FORMAT(14X,18HAVERAGE STRESSES (,3A4,52H) AT GAUSS POINTS OF REINFOUTS4350
     1ORCED CONCRETE ELEMENTS AT,1PE11.4,8H SECONDS//7X,7HELEMENT,5X,   OUTS4360
     25HNODES,5X,9HLOCATION ,  3(9X,6HSTRESS,9X,6HSTRAIN)/)             OUTS4370
 1750 CONTINUE                                                          OUTS4380
C                                                                       OUTS4390
C     PRINT OUT RESULTS.                                                OUTS4400
      WRITE(NPRT,6071) M,IP(M),IQ(M),(LOCA(I,K),I=1,3),                 OUTS4410
     1                 (STRESS(L),STRAIN(L),L=1,3)                      OUTS4420
 6071 FORMAT(10X,I2,5X,I2,4H TO ,I2,2X,3A4,1X,6(4X,1PE11.4))            OUTS4430
      IF (IL.LE.3) GO TO 1790                                           OUTS4440
      WRITE(NPRT,6072) (STRESS(L),STRAIN(L),L=4,IL)                     OUTS4450
 6072 FORMAT ((40X,6(4X,1PE11.4))/)                                     OUTS4460
 1790 CONTINUE                                                          OUTS4470
 1800 CONTINUE                                                          OUTS4480
```

```
C                                                                         OUTS4490
C     ******************************************************************  OUTS4500
C     *********    THIS SEGMENT REPORTS AVERAGE STRESSES AND    ********* OUTS4510
C     ********* STRAINS IN A NONCOMPOSITE WIDE FLANGE ELEMENT ********* OUTS4520
C     ******************************************************************  OUTS4530
C                                                                         OUTS4540
1999  LINE = NL                                                           OUTS4550
      DO 2100 M=1,NM                                                      OUTS4560
C                                                                         OUTS4570
C     VERIFY ELEMENT TYPE                                                 OUTS4580
      IF(MTYPE(M).NE.4) GO TO 2100                                        OUTS4590
      J = MATW(M)                                                         OUTS4600
      XLEN = XL(M)                                                        OUTS4610
      DO 2090 L=1,2                                                       OUTS4620
      IF(MSTAT(M).EQ.3) GO TO 2010                                        OUTS4630
C                                                                         OUTS4640
C     STRESS AND STRAIN COMPUTATIONS FOR AN ELASTIC ELEMENT               OUTS4650
      XLOC = 0.E0                                                         OUTS4660
      IF(L.EQ.2) XLOC = XLEN                                              OUTS4670
      CALL DEFO(M)                                                        OUTS4680
      DO 2000 K=1,2                                                       OUTS4690
      CALL STRN(M,XLOC,YFIBR(K*10-9,M),STRAIN(K))                         OUTS4700
      LSA(K) = ELA                                                        OUTS4710
      LSB(K) = TIC                                                        OUTS4720
      ELASMD = ET(J)                                                      OUTS4730
      IF(STRAIN(K).LE.0.E0) ELASMD = EC(J)                                OUTS4740
      STRESS(K) = ELASMD*STRAIN(K)                                        OUTS4750
      PCTYLD(K) = ABS(STRESS(K)/FCFY(J))                                  OUTS4760
      IF(PCTYLD(K).GE.1.E0) PCTYLD(K) = 1.E0                              OUTS4770
      PCTFX(K) = ABS(STRAIN(K)/STN(7,J))                                  OUTS4780
```

```
      IF(PCTFX(K).GE.1.E0) PCTFX(K) = 1.E0                              OUTS4790
 2000 CONTINUE                                                          OUTS4800
      GO TO 2070                                                        OUTS4810
C                                                                       OUTS4820
C     STRESS AND STRAIN DATA COLLECTION FOR YIELDED ELEMENTS            OUTS4830
C                                                                       OUTS4840
 2010 INDEX = KDATA(LPI+M)+21                                           OUTS4850
      STRAIN(1) = DATA(INDEX+(L-1)*11)                                  OUTS4860
      STRAIN(2) = DATA(INDEX+(L-1)*11+10)                               OUTS4870
C                                                                       OUTS4880
C     STRESSES AND FAILURE CRITERIA AT TOP AND BOTTOM FLANGES           OUTS4890
      INDEX = KDATA(LPSI+M) -1                                          OUTS4900
      DO 2060 K=1,2                                                     OUTS4910
      JJ = (K-1)*80+(L-1)*88+168                                        OUTS4920
      DO 2030 KK=1,8                                                    OUTS4930
      S(KK) = DATA(INDEX+JJ+KK)                                         OUTS4940
 2030 CONTINUE                                                          OUTS4950
      IF (S(1).LE.0.5E0) GO TO 2050                                     OUTS4960
      PCTYLD(K) = 1.E0                                                  OUTS4970
      PCTFX(K) = 1.E0                                                   OUTS4980
      STRESS(K) = 0.E0                                                  OUTS4990
      LSA(K) = RUP                                                      OUTS5000
      LSB(K) = URD                                                      OUTS5010
      GO TO 2060                                                        OUTS5020
 2050 POSNEG = -1.E0                                                    OUTS5030
      IF(STRAIN(K)-S(5).GE. 0.E0) POSNEG = 1.E0                         OUTS5040
      STRESS(K) = S(8)*POSNEG                                           OUTS5050
      LSB(K) = TIC                                                      OUTS5060
      LSA(K) = ELA                                                      OUTS5070
      IF(ABS(S(6)).GE.1.E-4) LSA(K) = PLA                               OUTS5080
```

```
      KM = INT(S(2))*6 + 2                                              OUTS5090
      PCTYLD(K) = S(8)/STS(KM,J)                                        OUTS5100
      IF(PCTYLD(K).GE.1.E0) PCTYLD(K) = 1.E0                            OUTS5110
      PCTFX(K) = (STRAIN(K)-S(5))/(POSNEG*STN(7,J))                     OUTS5120
      IF(PCTFX(K).GE.1.E0) PCTFX(K) = 1.E0                              OUTS5130
 2060 CONTINUE                                                          OUTS5140
C                                                                       OUTS5150
C     PRINTING SECTION                                                  OUTS5160
C                                                                       OUTS5170
2070   DO 2075 K=1,2                                                    OUTS5180
      SVSTRS((L-1)*2+K,M) = STRESS(K)                                   OUTS5190
      SVSTRN((L-1)*2+K,M) = STRAIN(K)                                   OUTS5200
2075   CONTINUE                                                         OUTS5210
      LINE = LINE + 3                                                   OUTS5220
      IF(LINE.LT.NL) GO TO 2080                                         OUTS5230
      CALL PAGE                                                         OUTS5240
      LINE = 7                                                          OUTS5250
      WRITE(NPRT,6210) (SUNIT(JS,NDIM),JS=1,3), TIME                    OUTS5260
 6210 FORMAT(13X,18HAVERAGE STRESSES (,3A4,                             OUTS5270
     *      38H) IN NONCOMPOSITE WIDE FLANGE ELEMENTS,                  OUTS5280
     2      3H AT,1PE11.4,8H SECONDS//21X,11HSITE IN THE,2X,2(9X,6HNORMALOUTS5290
     3      ),6X,7HLOADING,2(4X,7HPERCENT),3H OF/1X,3HEND,3X,7HELEMENT,6XOUTS5300
     4      ,13HCROSS SECTION,10X,6HSTRESS,9X,6HSTRAIN,7X,5HSTATE,4X,   OUTS5310
     5      9HOF  YIELD,4X,8HFRACTURE/1X,3H---,3X,7H-------,3X,7H-------,OUTS5320
     6      13H-------------,2(3X,12H------------),3X,7H-------,3X,     OUTS5330
     7      9H---------,3X,10H----------/)                              OUTS5340
 2080 IPQ = IP(M)                                                       OUTS5350
      IF(L.EQ.2) IPQ = IQ(M)                                            OUTS5360
      WRITE(NPRT,6220) IPQ,IP(M),IQ(M),(LABELS(K),K=1,3),STRESS(1),     OUTS5370
     2      STRAIN(1),LSA(1),LSB(1),PCTYLD(1),PCTFX(1),(LABELS(K),K=4,6),OUTS5380
```

```
     3      STRESS(2),STRAIN(2),LSA(2),LSB(2),PCTYLD(2),PCTFX(2)               OUTS5390
 6220 FORMAT(1X,I2,5X,I2,1H-,I2,8X,3A4, 7X,2(1PE11.4,4X),2A4,2X,               OUTS5400
     2      2(2PF8.3,5X)/21X,3A4, 7X,2(1PE11.4,4X),2A4,2X,2(2PF8.3,5X)/)      OUTS5410
 2090 CONTINUE                                                                 OUTS5420
 2100 CONTINUE                                                                 OUTS5430
      IF (IPRINT.LT.3) GO TO 2999                                              OUTS5440
C                                                                              OUTS5450
C     PRINT STRESSES AND STRAINS AT GAUSS POINTS FOR EXTENSIVE OUTPUT.        OUTS5460
      LINE = NL                                                                OUTS5470
      DO 2150 M=1,NM                                                           OUTS5480
C                                                                              OUTS5490
C     VERIFY ELEMENT TYPE.                                                     OUTS5500
      IF (MTYPE(M).NE.4) GO TO 2150                                            OUTS5510
      J = MATW(M)                                                              OUTS5520
      LINE = LINE + 8                                                          OUTS5530
      IF (LINE.LT.NL) GO TO 2180                                               OUTS5540
      CALL PAGE                                                                OUTS5550
      LINE = 8                                                                 OUTS5560
      WRITE(NPRT,6211) (SUNIT(JS,NDIM),JS=1,3), TIME                           OUTS5570
6211  FORMAT (11X,18HAVERAGE STRESSES (,3A4,44H) AT GAUSS POINTS OF WIDEOUTS5580
     1 FLANGE ELEMENTS AT,1PE11.4,8H SECONDS//7X,7HELEMENT,5X,6HNODES ,  OUTS5590
     2 3(9X,6HSTRESS,9X,6HSTRAIN)/)                                            OUTS5600
2180  CONTINUE                                                                 OUTS5610
      IF (MSTAT(M).EQ.3) GO TO 2186                                            OUTS5620
C                                                                              OUTS5630
C     STRESS AND STRAIN COMPUTATIONS FOR AN ELASTIC ELEMENT.                   OUTS5640
      XLEN = XL(M)                                                             OUTS5650
      CALL DEFO(M)                                                             OUTS5660
      IK = 0                                                                   OUTS5670
      DO 2184 I=1,7                                                            OUTS5680
```

```
      DO 2184 K=1,3                                                     OUTS5690
      IK = IK + 1                                                       OUTS5700
      CALL STRN(M,XPI(K,M),YGP(I,M),STRAIN(IK))                         OUTS5710
      ELASMD = ET(J)                                                    OUTS5720
      IF (STRAIN(IK).LE.0.E0) ELASMD = EC(J)                            OUTS5730
      STRESS(IK) = ELASMD*STRAIN(IK)                                    OUTS5740
 2184 CONTINUE                                                          OUTS5750
      GO TO 2194                                                        OUTS5760
C                                                                       OUTS5770
C     STRESS AND STRAIN DATA COLLECTION FOR YIELDED MEMBERS.            OUTS5780
 2186 CONTINUE                                                          OUTS5790
      KRESS = KDATA(LPSI+M)-1                                           OUTS5800
      KRAIN = KDATA(LPI+M) - 1                                          OUTS5810
      DO 2187 K=1,21                                                    OUTS5820
      STRAIN(K) = DATA(KRAIN+K)                                         OUTS5830
      STRESS(K) = DATA(KRESS+K*8)                                       OUTS5840
      IF(STRAIN(K).LT.DATA(KRESS+K*8-3))STRESS(K)= -STRESS(K)           OUTS5850
 2187 CONTINUE                                                          OUTS5860
 2194 CONTINUE                                                          OUTS5870
      WRITE(NPRT,6221) M,IP(M),IQ(M),(STRESS(K),STRAIN(K),K=1,21)       OUTS5880
 6221 FORMAT (10X,I2,5X,I2,4H TO ,I2,2X,7( 6 (4X,1PE11.4) /27X)/)       OUTS5890
 2150 CONTINUE                                                          OUTS5900
C                                                                       OUTS5910
  2999 IF(ISTRES.NE.2.AND.ISTRES.NE.3) GO TO 5000                       OUTS5920
C                                                                       OUTS5930
C     ****************************************************************** OUTS5940
C     **************** THIS SEGMENT REPORTS END FORCES **************** OUTS5950
C     **************** IN REINFORCED CONCRETE ELEMENTS **************** OUTS5960
C     ****************************************************************** OUTS5970
C                                                                       OUTS5980
```

```
      LINE = NL                                                         OUTS5990
      DO 3170 M=1,NM                                                    OUTS6000
C                                                                       OUTS6010
C     VERIFY ELEMENT TYPE                                               OUTS6020
      IF(MTYPE(M).GT.3) GO TO 3170                                      OUTS6030
C                                                                       OUTS6040
C     PRINTING SECTION                                                  OUTS6050
C                                                                       OUTS6060
      LINE = LINE + 5                                                   OUTS6070
      IF(LINE.LT.NL) GO TO 3160                                         OUTS6080
      CALL PAGE                                                         OUTS6090
      LINE = 9                                                          OUTS6100
      WRITE(NPRT,6300) TIME,(FUNIT(J,NDIM),J=1,4)                       OUTS6110
 6300 FORMAT(13X,46HREINFORCED CONCRETE ELEMENT INTERNAL FORCES AT,     OUTS6120
     2     1PE11.4,8H SECONDS//28X,8HUNITS - ,4A4//                     OUTS6130
     *                         17X,7HELEMENT,7X,6HSTRESS,10X,8HAT FIRST,OUTS6140
     3     8X,9HAT SECOND/17X,8HIDENTITY,5X,9HRESULTANT,2(7X,8HLISTED E,OUTS6150
     4     2HND)/17X,8H--------,5X,9H---------,1X,2(5X,12H------------)/)OUTS6160
 3160 WRITE(NPRT,6310) IP(M),IQ(M),SAVAXL(1,M),SAVAXL(2,M),SAVSHR(1,M), OUTS6170
     2 SAVSHR(2,M),SAVMOM(1,M),SAVMOM(2,M),SAVCRV(1,M),SAVCRV(2,M)      OUTS6180
 6310 FORMAT(18X,I2,1H-,I2,7X,5HAXIAL,4X,2(6X,1PE11.4)/30X,5HSHEAR,4X,  OUTS6190
     2     2(6X,1PE11.4)/30X,6HMOMENT,3X,2(6X,1PE11.4)/30X,9HCURVATURE, OUTS6200
     3     2(6X,1PE11.4)/)                                              OUTS6210
 3170 CONTINUE                                                          OUTS6220
C                                                                       OUTS6230
C     ******************************************************************OUTS6240
C     **************** THIS SEGMENT REPORTS END FORCES IN ***************OUTS6250
C     **************** NONCOMPOSITE WIDE FLANGE ELEMENTS  ***************OUTS6260
C     ******************************************************************OUTS6270
C                                                                       OUTS6280
```

```
      LINE = NL                                                         OUTS6290
      DO 4100 M=1,NM                                                    OUTS6300
C                                                                       OUTS6310
C     VERIFY ELEMENT TYPE                                               OUTS6320
      IF(MTYPE(M).NE.4) GO TO 4100                                      OUTS6330
C                                                                       OUTS6340
C     PRINTING SECTION                                                  OUTS6350
C                                                                       OUTS6360
      LINE = LINE + 5                                                   OUTS6370
      IF(LINE.LT.NL) GO TO 4090                                         OUTS6380
      CALL PAGE                                                         OUTS6390
      LINE = 9                                                          OUTS6400
      WRITE(NPRT,6400) TIME,(FUNIT(J,NDIM),J=1,4)                       OUTS6410
 6400 FORMAT(11X,51HNONCOMPOSITE WIDE FLANGE ELEMENT INTERNAL FORCES AT,OUTS6420
     2      1PE11.4,8H SECONDS//28X,8HUNITS - ,4A4//                    OUTS6430
     *                          17X,7HELEMENT,7X,6HSTRESS,10X,8HAT FIRST,OUTS6440
     3      8X,9HAT SECOND/17X,8HIDENTITY,5X,9HRESULTANT,2(7X,8HLISTED E,OUTS6450
     4     2HND)/17X,8H--------,5X,9H---------,1X,2(5X,12H------------)/)OUTS6460
 4090 WRITE(6,6410) IP(M),IQ(M),SAVAXL(1,M),SAVAXL(2,M),SAVSHR(1,M),    OUTS6470
     2 SAVSHR(2,M),SAVMOM(1,M),SAVMOM(2,M),SAVCRV(1,M),SAVCRV(2,M)      OUTS6480
 6410 FORMAT(18X,I2,1H-,I2,7X,5HAXIAL,4X,2(6X,1PE11.4)/30X,5HSHEAR,4X,  OUTS6490
     2      2(6X,1PE11.4)/30X,6HMOMENT,3X,2(6X,1PE11.4)/30X,9HCURVATURE, OUTS6500
     3      2(6X,1PE11.4)/)                                             OUTS6510
 4100 CONTINUE                                                          OUTS6520
C                                                                       OUTS6530
C     ***************************************************************** OUTS6540
C     **************** THIS SEGMENT REPORTS END FORCES **************** OUTS6550
C     ****************        IN LEAF SPRING ELEMENTS    **************** OUTS6560
C     ***************************************************************** OUTS6570
C                                                                       OUTS6580
```

```
      IF(NLS.EQ.0) GO TO 5000                                           OUTS6590
      LINE = NL                                                         OUTS6600
      DO 4510 L=1,NLS                                                   OUTS6610
      CALL LEAF(-L,UR,1)                                                OUTS6620
      SRQ(1) = -SRP(1)                                                  OUTS6630
      SRQ(2) = -SRP(2)                                                  OUTS6640
      DO 4480 I=1,3                                                     OUTS6650
      SAVSRP(I,L)=SRP(I)                                                OUTS6660
      SAVSRQ(I,L)=SRQ(I)                                                OUTS6670
 4480 CONTINUE                                                          OUTS6680
      LINE = LINE + 4                                                   OUTS6690
      IF(LINE.LT.NL) GO TO 4500                                         OUTS6700
      CALL PAGE                                                         OUTS6710
      LINE = 8                                                          OUTS6720
      WRITE(NPRT,6450) TIME,(FUNIT(J,NDIM),J=1,4)                       OUTS6730
 6450 FORMAT(17X,38HLEAF SPRING ELEMENT INTERNAL FORCES AT,1PE11.4,     OUTS6740
     2     8H SECONDS//28X,8HUNITS - ,4A4//                             OUTS6750
     *                    17X,7HELEMENT,7X,6HSTRESS,10X,8HAT FIRST,8X,  OUTS6760
     3     9HAT SECOND/17X,8HIDENTITY,5X,9HRESULTANT,2(7X,10HLISTED END)OUTS6770
     4     /17X,8H--------,5X,9H---------,1X,2(5X,12H------------)/)    OUTS6780
 4500 WRITE(NPRT,6460) IPL(L),IQL(L),SRP(1),SRQ(1),SRP(2),SRQ(2),       OUTS6790
     2     SRP(3),SRQ(3)                                                OUTS6800
 6460 FORMAT(18X,I2,1H-,I2,9X,5HAXIAL,2X,2(6X,1PE11.4)/32X,5HSHEAR,2X,  OUTS6810
     2     2(6X,1PE11.2)/32X,6HMOMENT,1X,2(6X,1PE11.4)/)                OUTS6820
 4510 CONTINUE                                                          OUTS6830
C                                                                       OUTS6840
C     ******************************************************************OUTS6850
C     *********** THIS SEGMENT REPORTS THE SOLUTION ACCURACY ***********OUTS6860
C     ******************************************************************OUTS6870
C                                                                       OUTS6880
```

```
 5000 CALL PAGE                                                            OUTS6890
C                                                                          OUTS6900
C      COMPUTE KINETIC ENERGY.                                             OUTS6910
       VALUES(5) = 0.E0                                                    OUTS6920
       IF (NMAS.EQ.0) GO TO 6501                                           OUTS6930
       DO 6500 I=1,NDFJ                                                    OUTS6940
       J = IDFI(I)                                                         OUTS6950
       K = IDFII(I)                                                        OUTS6960
6500   VALUES(5) = VALUES(5) + 0.5E0*DAS(K,J)*VEL(K,J)**2                  OUTS6970
C                                                                          OUTS6980
C      SYSTEM ENERGIES.                                                    OUTS6990
6501   IF(TIME.GT.TBEGIN) GO TO 6503                                       OUTS7000
       EBEGIN = VALUES(5)                                                  OUTS7010
       PBEGIN = -0.5E0*VALUES(2)                                           OUTS7020
       POTEN = 0.E0                                                        OUTS7030
       DO 6502 I=1,5                                                       OUTS7040
       DO 6502 J=1,4                                                       OUTS7050
6502   ENERGY(I,J) = 0.E0                                                  OUTS7060
6503   ENERGY(1,3) = VALUES(3)                                             OUTS7070
       ENERGY(2,3) = VALUES(4)                                             OUTS7080
       ENERGY(3,3) = VALUES(5)                                             OUTS7090
       ENERGY(4,3) = VALUES(3)+VALUES(4)+VALUES(5)                         OUTS7100
       ENERGY(5,3) = EBEGIN + POTEN - 0.5E0*VALUES(2)                      OUTS7110
       IF (IANAL.EQ.1) POTEN = PBEGIN                                      OUTS7120
       DO 6504 I=1,5                                                       OUTS7130
       ENERGY(I,2) = ENERGY(I,3) - ENERGY(I,1)                             OUTS7140
       IF(ENERGY(I,1).GT.TINY) ENERGY(I,4)=ENERGY(I,2)/ENERGY(I,1)*100.E0OUTS7150
       IF (ENERGY(I,2).EQ.0.E0) ENERGY(I,4) = 0.E0                         OUTS7160
6504   CONTINUE                                                            OUTS7170
       WRITE(NPRT,6508) TIME,(FUNIT(K,NDIM),K=1,4)                         OUTS7180
```

```
6508  FORMAT(29X,20HSOLUTION ACCURACY AT,1PE11.4,8H SECONDS//36X,       OUTS7190
     1   8HUNITS  - ,4A4///41X,15HSYSTEM ENERGIES//27X,7HINITIAL,10X,    OUTS7200
     2  6HCHANGE,10X,5HFINAL,7X,15HPER CENT CHANGE/10X,11(1H=),4(4X,     OUTS7210
     3  12(1H=))/)                                                       OUTS7220
      DO 6510 I=1,5                                                      OUTS7230
      IF (ENERGY(I,1).GT.TINY.OR.ENERGY(I,2).EQ.0.E0)                    OUTS7240
     1     WRITE(NPRT,6512) (ETYPE(K,I),K=1,3),(ENERGY(I,J),J=1,4)       OUTS7250
      IF (ENERGY(I,1).LE.TINY.AND.ENERGY(I,2).NE.0.E0)                   OUTS7260
     1     WRITE(NPRT,6514) (ETYPE(K,I),K=1,3),(ENERGY(I,J),J=1,3)       OUTS7270
      IF (I.EQ.3) WRITE (NPRT,6515)                                      OUTS7280
      IF (I.EQ.4) WRITE (NPRT,6516)                                      OUTS7290
6510  CONTINUE                                                           OUTS7300
      WRITE (NPRT,6516)                                                  OUTS7310
6512  FORMAT (10X,3A4,3X,3(1PE12.4,4X),0PG12.4)                          OUTS7320
6514  FORMAT (10X,3A4,3X,3(1PE12.4,4X),2X,8HINFINITE)                    OUTS7330
6515  FORMAT (21X,4(4X,12(1H-)))                                         OUTS7340
6516  FORMAT (21X,4(4X,12(1H=)))                                         OUTS7350
      DO 6518 I=1,5                                                      OUTS7360
6518  ENERGY(I,1) = ENERGY(I,3)                                          OUTS7370
      LINE = 25                                                          OUTS7380
      WRITE (NPRT,6520)                                                  OUTS7390
6520  FORMAT(//8X,9HJOINT AND,7X,8HVALUE OF,7X,8HRESIDUAL,9X,8HRESIDUAL,OUTS7400
     2      6X,11HREL. ENERGY/8X,9HDIRECTION,5X,12HIMPOSED LOAD,4X,      OUTS7410
     3      11HFORCE,  MAX.,5X,12HENERGY,  MAX.,7X,5HERROR/8X,9H---------, OUTS7420
     4      2(5X,12H------------,4X,11H-----------)/)                   OUTS7430
      DO 5020 J=1,NJ                                                     OUTS7440
      LINE = LINE + 4                                                    OUTS7450
      IF(LINE.LT.NL) GO TO 5010                                          OUTS7460
      CALL PAGE                                                          OUTS7470
      LINE = 8                                                           OUTS7480
```

```
      WRITE(NPRT,6520)                                                  OUTS7490
 5010 DO 5015 I=1,3                                                     OUTS7500
      DMNSLZ=AVGL                                                       OUTS7510
      IF(I.EQ.3) DMNSLZ=1.E0                                            OUTS7520
      F(I,J) = F(I,J)/DMNSLZ                                            OUTS7530
      WRITE(NPRT,6530)    J,AD(I),F(I,J),DER(I,J),RESENG(I,J),ERJF(I,J) OUTS7540
 6530 FORMAT(1H ,  9X,I2,3X,A2,1X,4(5X,1PE11.4))                        OUTS7550
      F(I,J)=F(I,J)*DMNSLZ                                              OUTS7560
 5015 CONTINUE                                                          OUTS7570
      WRITE(NPRT,6020)                                                  OUTS7580
 5020 CONTINUE                                                          OUTS7590
      RETURN                                                            OUTS7600
      END                                                               OUTS7610
```

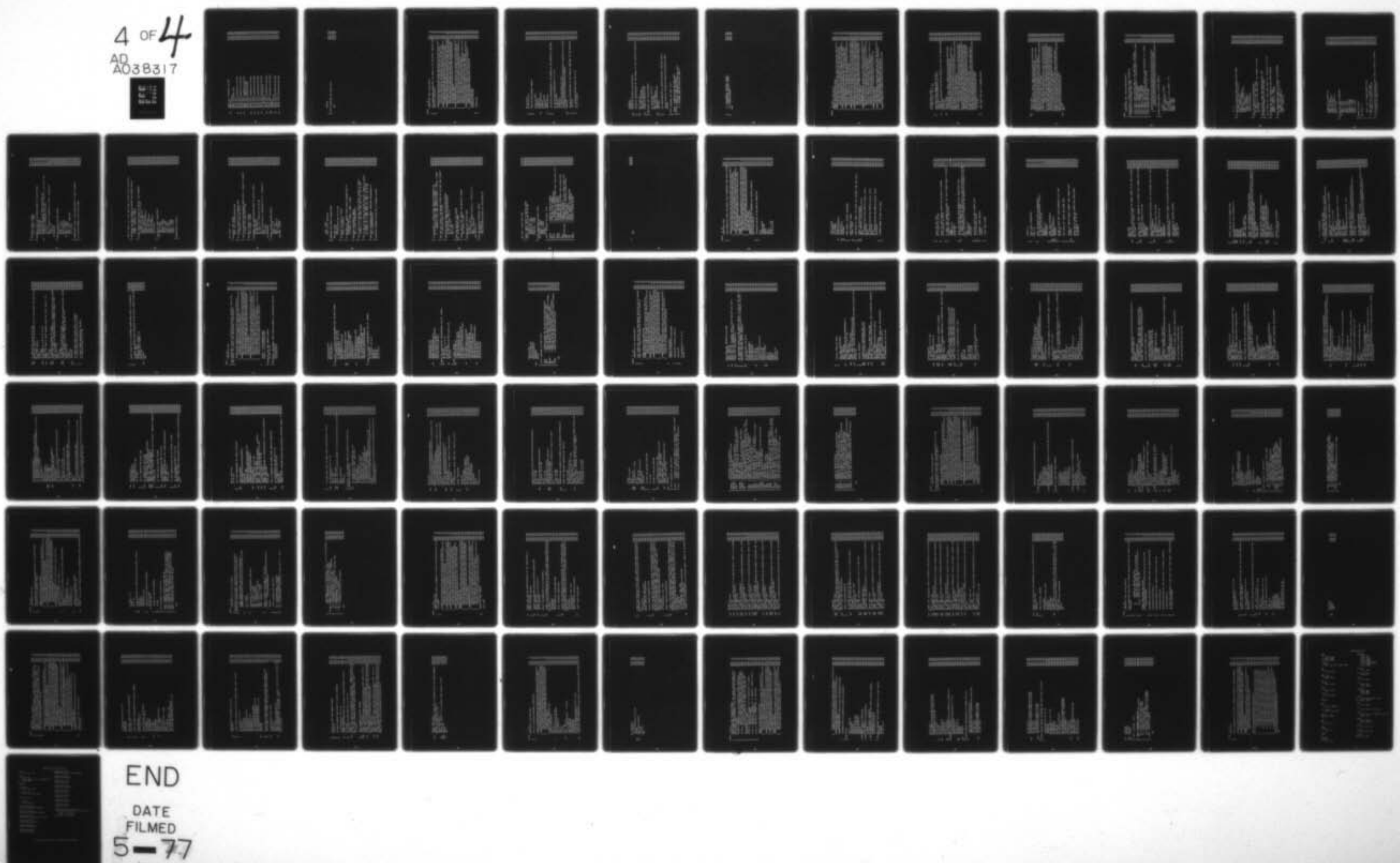
```
CPAGE    0  10
      SUBROUTINE PAGE
C
C     THIS SUBROUTINE PRINTS PROBLEM TITLE AND PAGE NO.
C
      COMMON/LEADBK/IDUM1(29),HEAD(20),IDUM2(11)
      COMMON/MAINBK/JD1(6),IPAGE,JD2(10),LINE,JD3(24),NPRT,JD4(6)
      COMMON/PRNTBK/MPRINT
      INTEGER HEAD
C
      IF (MPRINT.EQ.0.AND.IPAGE.GT.0) GO TO 30
      IPAGE = IPAGE + 1
20    FORMAT (1H1,5X,20A4,25X,4HPAGE,I5//)
      WRITE (NPRT,20) HEAD,IPAGE
      LINE = 12
      RETURN
30    WRITE (NPRT,40)
40    FORMAT (1H0)
      LINE = 3
      RETURN
      END
```

```
CPLOG    0  10
      SUBROUTINE PLOG(IT)
C
C THIS SUBROUTINE SAVES THE PLOT FILE.
C
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20),
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,
     4    NTIMES,NVEL,IINITD
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45),
     5    YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)
      COMMON /PLOTBK/ ITER
      COMMON/SAVEBK/SAVACC(3,50),SAVAXL(2,45),SAVCRV(2,45),SAVMOM(2,45)
     1    ,SAVSHR(2,45),SAVSRP(3,20),SAVSRQ(3,20),SAVXDJ(3,50),
     2    SAVVEL(3,50),SVSTRN(12,45),SVSTRS(12,45)
      INTEGER HEAD,DHEAD
      IF (ITER.GT.0) GO TO 20
      REWIND IT
      WRITE (IT) (HEAD(I),I=1,20)
      WRITE(IT) IUNITS,NDF,NJ,NM,NLS,(IP(M),IQ(M),M=1,NM)
```

AD-A038 317 VIRGINIA POLYTECHNIC INST AND STATE UNIV BLACKSBURG --ETC F/G 13/13
RELIABILITY STUDY OF SINGER. VOLUME II. USER'S MANUAL.(U)
JAN 77 A E SOMERS, S M HOLZER, J C BRADSHAW F29601-75-C-0050

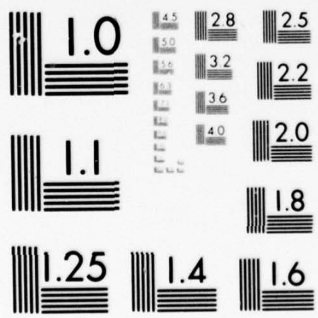
UNCLASSIFIED AFWL-TR-76-192-VOL-2 NL

4 OF 4
AD
A038317



END
DATE
FILMED
5-77

38317



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
      WRITE (IT) (MTYPE(M),M=1,NM)
20    CONTINUE
      ITER = ITER+1
      WRITE (IT) ITER,TIME
      DO 21 I=1,3
21    WRITE (IT) (SAVXDJ(I,J),J=1,NJ)
      DO 22 I=1,3
22    WRITE (IT) (SAVACC(I,J),J=1,NJ)
      DO 23 I=1,3
23    WRITE (IT) (SAVVEL(I,J),J=1,NJ)
      WRITE (IT) (URM(M),M=1,NM)
      WRITE (IT) (UDM(M),M=1,NM)
      DO 24 I=1,2
24    WRITE (IT) (SAVCRV(I,M),M=1,NM)
      DO 25 I=1,2
25    WRITE (IT) (SAVMOM(I,M),M=1,NM)
      DO 26 I=1,2
26    WRITE (IT) (SAVSHR(I,M),M=1,NM)
      DO 27 I=1,2
27    WRITE (IT) (SAVAXL(I,M),M=1,NM)
      IF (NLS.EQ.0) GO TO 30
      DO 28 I=1,3
28    WRITE (IT) (SAVSRP(I,M),M=1,NLS)
      DO 29 I=1,3
29    WRITE (IT) (SAVSRQ(I,M),M=1,NLS)
30    CONTINUE
      DO 31 I=1,12
31    WRITE (IT) (SVSTRS(I,M),M=1,NM)
      DO 32 I=1,12
32    WRITE (IT) (SVSTRN(I,M),M=1,NM)
```

```
      RETURN
C
C ***** GLOSSARY FOR PLOG *****
C
      END
```

```
CPOTE    0  10
      SUBROUTINE POTE (SOLN,VALUE)                                      POTE   0
C                                                                       POTE  10
C     THIS SUBROUTINE CALCULATES THE VALUE OF THE POTENTIAL FUNCTION FORPOTE  20
C     SPECIFIED VARIABLES.                                              POTE  30
C                                                                       POTE  40
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), POTE  50
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),      POTE  60
     1    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)     POTE  70
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), POTE  80
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      POTE  90
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,POTE 100
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,POTE 110
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,MINC,NJ,NJD,NJER,NL,NLD, POTE 120
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  POTE 130
     4    NTIMES,NVEL,IINITD                                            POTE 140
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),POTE 150
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),    POTE 160
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),POTE 170
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                POTE 180
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45), POTE 190
     5    YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)              POTE 200
      COMMON/SEEKBK/DEFOR(90),STPSIZ(90),GRAD(90),GRADI(90),DELTAG(90), POTE 210
     1    DIRECT(90),DIAG(90),STEP(4),DSTEP(4),FVAL(4),VALUES(7),       POTE 220
     2    DISACC,SSIZE,FUNACC,FUNMIN,CRITL,CRITU,NLIN                   POTE 230
      INTEGER HEAD,DHEAD                                                POTE 240
      DIMENSION SOLN(90)                                                POTE 250
C                                                                       POTE 260
C     INITIALIZE.                                                       POTE 270
C                                                                       POTE 280
```

```
      DO 11 I=1,4                                                       POTE 290
11    VALUES(I)=0.E0                                                    POTE 300
C                                                                       POTE 310
C     FORM 2D DISPLACEMENT ARRAYS.                                      POTE 320
C                                                                       POTE 330
      DO 20 I=1,NDFJ                                                    POTE 340
      J=IDFI(I)                                                         POTE 350
      K=IDFII(I)                                                        POTE 360
20    XDJ(K,J)=SOLN(I)                                                  POTE 370
      L=NDFJ+1                                                          POTE 380
      DO 21 I=L,NDF                                                     POTE 390
      M=IDFI(I)                                                         POTE 400
21    XDM(M)=SOLN(I)                                                    POTE 410
C                                                                       POTE 420
C     CALCULATE CONTRIBUTIONS OF THE MASSES TO THE POTENTIAL FUNCTION.  POTE 430
C                                                                       POTE 440
      IF (DT.EQ.0.E0.OR.NMAS.EQ.0) GO TO 40                             POTE 450
      DO 30 I=1,NDFJ                                                    POTE 460
      J=IDFI(I)                                                         POTE 470
      K=IDFII(I)                                                        POTE 480
      IF (DAS(K,J).EQ.0.E0) GO TO 30                                    POTE 490
      VALUES(1)=VALUES(1)+DAS(K,J)*((3.E0*SOLN(I)-6.E0*DIS(K,J))/(DT*DT)POTE 500
     1-6.E0*VEL(K,J)/DT-2.E0*ACC(K,J))*SOLN(I)                          POTE 510
30    CONTINUE                                                          POTE 520
C                                                                       POTE 530
C     CALCULATE CONTRIBUTIONS OF THE FORCING FUNCTIONS TO THE POTENTIAL POTE 540
C     FUNCTION.                                                         POTE 550
C                                                                       POTE 560
40    IF (IFOR.EQ.0.AND.NFF.EQ.0) GO TO 60                              POTE 570
      DO 50 I=1,NDFJ                                                    POTE 580
```

```
      J=IDFI(I)
      K=IDFII(I)
50    VALUES(2)=VALUES(2)-F(K,J)*SOLN(I)
C
C     CALCULATE CONTRIBUTIONS OF THE MEMBERS TO THE POTENTIAL FUNCTION.
C
60    DO 70 M=1,NM
      CALL ENGY (M,UR,UD)
      VALUES(3)=VALUES(3)+UR
70    VALUES(4)=VALUES(4)+UD
C
C CALCULATE LEAF SPRING ENERGY.
C
      IF (NLS.LE.0) GO TO 74
      DO 72 M=1,NLS
      CALL LEAF(-M,UR,1)
72    VALUES(3)=VALUES(3)+UR
74    CONTINUE
C
C     SUM CONTRIBUTIONS TO OBTAIN POTENTIAL FUNCTION VALUE.
C
      VALUE=(VALUES(1)+VALUES(2)+VALUES(3)+VALUES(4))/AVDM
      RETURN
C
C ***** GLOSSARY FOR POTE *****
C
C  VALUE = FUNCTION VALUE OR TOTAL WORK.
C VALUES = ENERGY OR WORK COMPONENTS.
C      1 = INERTIAL COMPONENTS OF EXTERNAL WORK.
C      2 = EXTERNAL WORK OR POTENTIAL ENERGY.
```

```
C       3 = RECOVERABLE INTERNAL ENERGY.                               POTE 890
C       4 = DISSIPATIVE INTERNAL ENERGY.                               POTE 900
C       5 = KINETIC ENERGY.                                            POTE 910
      END                                                              POTE 920
```

```
CREGO    0  10
      SUBROUTINE REGO(NFILE)                                            REGO   0
      COMMON DATA(10000),KDATA(500)                                     REGO  10
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), REGO  20
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),      REGO  30
     2    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)     REGO  40
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), REGO  50
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      REGO  60
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ  (45),IQL(20),MATR(45),     REGO  70
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),REGO  80
     2    MTYPE(45 ),NGRP(45),NSPAC(6,45),NTIES(45)                     REGO  90
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9), REGO 100
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)REGO 110
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,REGO 120
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,REGO 130
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, REGO 140
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  REGO 150
     4    NTIMES,NVEL,IINITD                                            REGO 160
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),REGO 170
     1   BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),     REGO 180
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),REGO 190
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                REGO 200
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45), REGO 210
     5    YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)              REGO 220
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,          REGO 230
     1      LTABI,NMAX,NMAXI                                            REGO 240
      COMMON/SEEKBK/DEFOR(90),STPSIZ(90),GRAD(90),GRADI(90),DELTAG(90), REGO 250
     1      DIRECT(90),DIAG(90),STEP(4),DSTEP(4),FVAL(4),VALUES(7),     REGO 260
     2      DISACC,SSIZE,FUNACC,FUNMIN,CRITL,CRITU,NLIN                 REGO 270
      COMMON/STRNBK/SRP(4),SRQ(4),UX,UY,UZ,XLEN,AREA,ZZI,IMAT           REGO 280
```

```
      INTEGER HEAD, DHEAD                                               REGO 290
      IF (NFILE.GT.0) GO TO 100                                         REGO 300
C                                                                       REGO 310
C     MOVE INFO FROM COMMON BLOCK TO UNIT                               REGO 320
      IF(LERR.EQ.0.AND.IERR.EQ.0) GO TO 20                              REGO 330
      PRINT 15                                                          REGO 340
15    FORMAT(1H ,62H***NO FILE WRITTEN BECAUSE OF INPUT OR STORAGE ERRORREGO 350
     1S(REGO)***)                                                       REGO 360
      GO TO 200                                                         REGO 370
20    REWIND NSAVE                                                      REGO 380
      WRITE(NSAVE) DATA, KDATA, ICARD,IP,IPL,IQ,IQL,MATR,MATW,MBAR,     REGO 390
     1 MCODE,MSHEAR,MSTAT,MTIES,MTYPE,NGRP,NSPAC,NTIES,                 REGO 400
     2DENS,EC,EPSU,ST,FCFY,G,PR,S,SLOPE,ST,STN,STS,UNLK,ICODE,NAME,     REGO 410
     3DAS,DIS,ERJF,ERJH,ERJZ,F,FOR,VEL,X,XDJ,Y,IDFI,IDFII,              REGO 420
     4AVDM,AVG ,CA,CB,CC,CD,CE,DHEAD,DT,EPS,HEAD,PI,RERF,RERH,RERZ,SERR,REGO 430
     4TBEGIN,THALT,TIME,TINK,TINY,TPROB,                                REGO 440
     5IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,IREC,ISTART,  REGO 450
     6       ISTOP,ISTRES ,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,NCRD,   REGO 460
     7NOF,NOFD,NOFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,NLS,NLSR, REGO 470
     8NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,NTIMES,NVEL    REGO 480
      WRITE(NSAVE)BETA,BMEM,BPP,BWF,D,DP,DPP,DWF,EFFL,EFLM,HMEM,HTOP,   REGO 490
     AHTWF,PDP,SPRING,STIES,TFWF,TWWF,UDM,URM,WFPI,XBEG,XBEGM,XBEGS,    REGO 500
     BXL,XPI,YBAR,YFIBR,YLDS,ZI,AGRP,ATIES,                             REGO 510
     DLCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,LTABI,NMAX,NMAXI,      REGO 520
     EDEFOR,STPSIZ,GRAD,GRADI,DELTAG,DIRECT,DIAG,STEP,DSTEP,FVAL,DISACC,REGO 530
     ESSIZE,FUNACC,FUNMIN,CRITL,CRITU,NLIN,                             REGO 540
     FSRP,SRQ,UX,UY,UZ,XLEN,AREA,ZZI,IMAT                               REGO 550
       GO TO 200                                                        REGO 560
C                                                                       REGO 570
C      MOVE INFO FROM UNIT TO COMMON BLOCKS                             REGO 580
```

```
C                                                                       REGO 590
100    REWIND NTAPE                                                     REGO 600
        READ(NTAPE) DATA, KDATA, ICARD,IP,IPL,IQ,IQL,MATR,MATW,MBAR,    REGO 610
      1 MCODE,MSHEAR,MSTAT,MTIES,MTYPE,NGRP,NSPAC,NTIES,                REGO 620
      2DENS,EC,EPSU,ST,FCFY,G,PR,S,SLOPE,ST,STN,STS,UNLK,ICODE,NAME,    REGO 630
      3DAS,DIS,ERJF,ERJH,ERJZ,F,FOR,VEL,X,XDJ,Y,IDFI,IDFII,             REGO 640
      4AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD,DT,EPS,HEAD,PI,RERF,RERH,RERZ,SERR,REGO 650
      4TBEGIN,THALT,TIME,TINK,TINY,TPROB,                               REGO 660
      5IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,IREC,ISTART, REGO 670
      6       ISTOP,ISTRES ,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,NCRD,  REGO 680
      7NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,NLS,NLSR, REGO 690
      8NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,NTIMES,NVEL   REGO 700
       READ(NTAPE)BETA,BMEM,BPP,BWF,D,DP,DPP,DWF,EFFL,EFLM,HMEM,HTOP,   REGO 710
      AHTWF,PDP,SPRING,STIES,TFWF,TWWF,UDM,URM,WFPI,XBEG,XBEGM,XBEGS,   REGO 720
      BXL,XPI,YBAR,YFIBR,YLDS,ZI,AGRP,ATIES,                            REGO 730
      DLCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,LTABI,NMAX,NMAXI,     REGO 740
      EDEFOR,STPSIZ,GRAD,GRADI,DELTAG,DIRECT,DIAG,STEP,DSTEP,FVAL,DISACC,REGO 750
      ESSIZE,FUNACC,FUNMIN,CRITL,CRITU,NLIN,                            REGO 760
      FSRP,SRQ,UX,UY,UZ,XLEN,AREA,ZZI,IMAT                              REGO 770
200    RETURN                                                           REGO 780
       END                                                              REGO 790
```

```
CREIN    0  10
      SUBROUTINE REIN (MATL,RAINX,URS,UDS)
C
C DEFINITION OF STRESS HISTORY CURVE PARAMETERS
C   TRANSMITTED VIA COMMON ARRAY S(9)
C
C   S(1)= XFAIL= FIBER FAILURE CODE; 0= NOT RUPTURED,1= RUPTURED
C   S(2)= RELD = RELOADING INDICATOR; 0= VIRGIN CURVE,1= RELOADING CURVE
C   S(3)= RMAX = MAXIMUM STRAIN
C   S(4)= SMAX = MAXIMUM STRESS
C   S(5)= ZRAINX = STRAIN ORIGIN
C   S(6)= PLAS = PLASTIC OFFSET
C   S(7)= SSEGM= LINE SEGMENT OF MATERIAL CURVE CONTAINING LAST STRAIN
C   S(8)= SRESX= ABSOLUTE VALUE OF LAST STRESS
C   S(9)= SRAINX=LAST STRAIN
C
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),
     $      XFAIL,RELD,RMAX,SMAX,ZRAINX,PLAS,SSEGM,SRESX,SRAINX,
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)
C
C DEFINE STRAIN ORIGIN ^ ELASTIC MODULUS
C
      J= MATL
      ZERON = STN(1,J)+ ZRAINX
      YIT = SLOPE(1,J)
      YIC = YIT
      IF (SSEGM .LT. 1.E0) SSEGM = 1.E0
      SEGM = SSEGM
      PRMAX = RMAX
      PSMAX = SMAX
```

```
      N = 7
C
C IF LOAD HAS RECYCLED ADJUST STRESS-STRAIN CURVE
      IF (RELD .LT. 1.E0 ) GO TO 2
      SAVE = SLOPE(2,J)
      SLOPE(2,J)= SLOPE(8,J)
      SLOPE (1,J) = SLOPE (7,J)
      SAVN = STN(2,J)
       STN(2,J)=STN(8,J)
       SAVS = STS(2,J)
       STS(2,J)= STS(8,J)
C
C TEST FOR TENSION OR COMPRESSION
    2 IF (SMAX.LT. 0.E0 ) GO TO 200
      IF (RELD .LT. 1.E0 .AND. RAINX .LE. ZERON) GO TO 200
      GO TO 100
C
C TENSILE CURVE --COME FROM PREVIOUS COMP LOAD
C IF ELASTIC MEMBER DO NOT ALTER CURVE
C
   90 IF (RELD .LT. 1.E0 .AND. SMAX .GT. -STS(2,J)) GO TO 95
C
C DETERMINE NEW ZERO STRAIN AND NEW STRAIN ORIGIN.
      ZRAINX = ZRAINX + PLAS
      ZERON = STN(1,J) + ZRAINX
C
C ALTER STRESS-STRAIN CURVE IF INITIAL LOAD REVERSAL
      IF (RELD .EQ. 1.E0 ) GO TO  95
      SLOPE(1,J)= SLOPE(7,J)
      SAVE = SLOPE(2,J)
```

```
      SLOPE(2,J)= SLOPE(8,J)
      SAVN= STN(2,J)
      STN(2,J)= STN(8,J)
      SAVS= STS(2,J)
      STS(2,J)= STS(8,J)
      RELD = 1.E0
C
C INITIALIZE CURVE PARAMETERS
   95 SMAX = 0.E0
      RMAX = 0.E0
      PLAS = 0.E0
      SEGM = 1.E0
      PSMAX = 0.E0
      SRAINX= ZERON
      PRMAX = ZERON
      SRESX = 0.E0
      SSEGM = 1.E0
C
C TENSILE LOADING
C
  100 ISIGN = 1
C
C
      RESX = YIT * (RAINX-ZERON -PLAS)
C
C IF COMP STRESS GO TO COMP CURVE
      IF (RESX .LT. 0.E0) GO TO 190
C
C TEST FOR UNLOADING OR RELOADING (KEEP STRESS ^ BRANCH TO ENERGY)
      IF (RESX .LT. SMAX) GO TO 300
```

```
C
C CALCULATE EFFECT STRAIN
      STZ = RAINX - ZERON
C
C INCREMENT THRU STRESS-STRAIN CURVE TO LOCATE STRESS
      IPT=SEGM+1.1E0
      DO 50 I=IPT,N
      IF(I.GT.8.OR.J.GT.9) PRINT 49,SEGM,I,J,N
49    FORMAT(1H ,7HSEGMENT,E14.7,4H I= ,I2,4H J= ,I2,4H N= ,I2)
      IF (STZ.GT. STN(I,J)) GO TO 50
      II= I-1
      RESX = STS(II,J)+ SLOPE(II,J) *(STZ - STN(II,J))
       SEGM = II
      GO TO 60
   50 CONTINUE
C
C STRAIN EXCEEDS RUPTURE STRAIN
      SEGM = N-1
      XFAIL = 1.E0
      RESX= STS(7,J)
      STZ= STN(N,J)
   60 RMAX = RAINX
      SMAX = RESX
      PLAS = STZ - RESX/YIT
      GO TO 300
C
C COMPRESSIVE CURVE --COME FROM PREVIOUS TENSILE LOADING
C
C IF ELASTIC MEMBER DO NOT ALTER CURVE
C
```

```
  190 IF (RELD .LT. 1.E0 .AND. SMAX .LT.STS (2,J)) GO TO 195
C
C DETERMINE NEW ZERO STRAIN AND DEFINE NEW STRAIN ORIGIN.
      ZRAINX = ZRAINX + PLAS
      ZERON = STN(1,J)+ ZRAINX
C
C ALTER STRESS-STRAIN CURVE IF INITIAL LOAD REVERSAL
      IF (RELD .EQ. 1.E0 ) GO TO 195
      SLOPE(1,J)= SLOPE(7,J)
      SAVE = SLOPE(2,J)
      SLOPE(2,J)= SLOPE(8,J)
      SAVS= STS(2,J)
      STS(2,J) = STS(8,J)
      SAVN= STN(2,J)
      STN(2,J)= STN(8,J)
      RELD =1.E0
C
C INITIALIZE CURVE PARAMETERS
  195 SMAX =0.E0
      RMAX = 0.E0
      SRAINX= ZERON
      SRESX = 0.E0
       PLAS =0.E0
      SEGM =1.E0
      PSMAX = 0.E0
      PRMAX = ZERON
      SSEGM = 1.E0
C
C COMPRESSIVE LOADING
C
```

```
  200 ISIGN = -1                                                        REIN1490
C                                                                       REIN1500
C CALCULATE TEST STRESS                                                 REIN1510
      RESX= YIC *(RAINX-ZERON-PLAS)                                     REIN1520
C                                                                       REIN1530
C IF TENSILE STRESS GO TO TENSION CURVE                                 REIN1540
      IF (RESX.GT. 0.E0) GO TO 90                                       REIN1550
C                                                                       REIN1560
C TEST FOR UNLOADING OR RELOADING (KEEP STRESS ^ BRANCH TO ENERGY)      REIN1570
      IF (RESX .GT. SMAX) GO TO 300                                     REIN1580
C                                                                       REIN1590
C CALCULATE EFFECT STRAIN                                               REIN1600
      STZ = RAINX - ZERON                                               REIN1610
C                                                                       REIN1620
C INCREMENT THRU STRESS-STRAIN CURVE TO LOCATE STRESS                   REIN1630
      IPT=SEGM+1.1E0                                                    REIN1640
      DO 150 I=IPT,N                                                    REIN1650
      IF (STZ .LT. -STN(I,J)) GO TO 150                                 REIN1660
      II= I-1                                                           REIN1670
      RESX = -STS(II,J) + SLOPE(II,J)*( STZ + STN(II,J))                REIN1680
      SEGM =II                                                          REIN1690
      GO TO 160                                                         REIN1700
  150 CONTINUE                                                          REIN1710
C                                                                       REIN1720
C STRAIN EXCEEDS RUPTURE STRAIN                                         REIN1730
      XFAIL = 1.E0                                                      REIN1740
      SEGM = N-1                                                        REIN1750
      RESX= -STS(7,J)                                                   REIN1760
      STZ= STN(N,J)                                                     REIN1770
  160 RMAX = RAINX                                                      REIN1780
```

```
      SMAX = RESX
      PLAS = STZ - RESX/YIC
C
C PROCEED WITH ENERGY CALCULATIONS
C
  300 URS= 0.E0
      UDS= 0.E0
      IF(XFAIL.GT.0.E0) GO TO 400
C
C CALCULATE ABS VALUE OF STRESS
      ARESX = ABS(RESX)
C
C IF RELOADING TO  MASTER CURVE LOCATE INTERSECTING PT
      PSMAX=ABS(PSMAX)
      IF (SRESX .LT. PSMAX ) SRESX = PSMAX
C
C CALCULATE RECOVERABLE ENERGY
      URS = 0.5E0 *RESX*RESX/YIT
C
C TEST TO SEE IF ON INITIAL ELASTIC CURVE- BRANCH TO ENERGY
      IF(RELD .LT. 1.E0 .AND. SEGM .EQ. 1.E0 ) GO TO 400
C
C TEST TO SEE IF ON UNLOAD OR RELOAD SECTION - BRANCH TO ENERGY
      IF (ISIGN .GT. 0 .AND. RESX .LT. SMAX ) GO TO 400
      IF (ISIGN .LT. 0 .AND. RESX .GT. SMAX ) GO TO 400
C
C CALCULATE DISSIPATED ENERGY (PREVIOUS RECOVERABLE)*2
      UDS = PSMAX*PSMAX/YIT
C
C CALCULATE INCREASE IN UDS DUE TO CHANGED LOADING
```

```
C
C IF RELOADING TO  MASTER CURVE LOCATE INTERSECTING PT
      IF (( SRAINX.GT.0.E0 .AND. SRAINX.LT.PRMAX ) .OR.
     1    ( SRAINX.LT.0.E0 .AND. SRAINX.GT.PRMAX )) SRAINX = PRMAX
C
C ARE THE LAST STRAIN ^ THE PRESENT STRAIN IN THE SAME LINE SEGMENT
      IF (SSEGM.LT.SEGM) GO TO 330
C
C CALCULATE UDS FOR POINTS IN THE SAME SEGMENT
      R =ABS((ARESX + SRESX)*(RAINX-SRAINX))
      IF (R .LT. 0.E0 ) R= -R
      UDS = UDS + R
      GO TO 360
C
C FIND ABS VALUE OF LAST STRAIN
330   R=ABS(SRAINX-ZERON)
C
C LOCATE END OF LINE SEGMENT FOR PREVIOUS LOADING
      IPT= SSEGM + 1.1E0
C
C ACCUMULATE AREA UNDER CURVE
      N= SEGM
      DO 340 I= IPT,N
      UDS  = UDS + (SRESX+ STS(I,J)) * (STN(I,J) -R)
      SRESX = STS(I,J)
  340 R=STN(I,J)
C
C USE ABS VALUE OF EFFECTIVE STRAIN
      STZ=ABS(STZ)
      UDS= UDS+(ARESX+STS(N,J))* (STZ-STN(N,J))
```

```
  360 UDS = 0.5E0 *UDS -URS                                           REIN2390
C                                                                     REIN2400
C CHECK ^ RESET CURVE TO INITIAL SHAPE                                REIN2410
  400 IF (RELD .LT. 1.E0) GO TO 450                                   REIN2420
      SLOPE(1,J)= YIT                                                 REIN2430
      SLOPE(2,J)= SAVE                                                REIN2440
      STS(2,J)  = SAVS                                                REIN2450
      STN(2,J)  = SAVN                                                REIN2460
C                                                                     REIN2470
C STORE PARAMETERS FOR NEXT CYCLE                                     REIN2480
  450 SRAINX = RAINX                                                  REIN2490
      SRESX = ARESX                                                   REIN2500
      SSEGM = SEGM                                                    REIN2510
      RETURN                                                          REIN2520
C *****  GLOSSARY FOR REIN   *****                                    REIN2530
C                                                                     REIN2540
C ARESX          = ABSOLUTE VALUE OF STRESS.                          REIN2550
C IPT            = POINT ON A LINE SEGMENT.                           REIN2560
C ISIGN          = 1, TENSION ; -1, COMPRESSION.                      REIN2570
C N              = NO OF POINTS IN STRESS-STRAIN CJRVE (FOR STEEL N=7). REIN2580
C PRMAX          = PREVIOUS MAXIMUM STRAIN.                           REIN2590
C PSMAX          = PREVIOUS MAXIMUM STRESS.                           REIN2600
C R              = INTERMEDIATE STEP IN ENERGY AREA CALCULATION.      REIN2610
C RESX           = STRESS IN QUESTION.                                REIN2620
C SAVE,SAVN,SAVS = TEMPORARY STORAGE FOR YIELD POINT CURVE DATA.      REIN2630
C SEGM           = LINE SEGMENT NO OF PREVIOUS STRAIN STATE.          REIN2640
C STZ            = EFFECTIVE STRAIN.                                  REIN2650
C YIC            = YOUNG+S MODULUS IN COMPRESSION.                    REIN2660
C YIT            = YOUNG+S MODULUS IN TENSION.                        REIN2670
C ZERON          = EFFECTIVE STRAIN ORIGIN.                           REIN2680
```

```
C                                                                  REIN2690
      END                                                          REIN2700
```

```
CREJO    0  10
      SUBROUTINE REJO                                                   REJO   0
C                                                                       REJO  10
C     THIS SUBROUTINE READS AND CHECKS JOINT COORDINATE AND JOINT RESTRAREJO  20
C                                                                       REJO  30
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), REJO  40
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),      REJO  50
     1    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)     REJO  60
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), REJO  70
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      REJO  80
      INTEGER HEAD,DHEAD                                                REJO  90
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,REJO 100
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,REJO 110
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, REJO 120
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  REJO 130
     4    NTIMES,NVEL,IINITD                                            REJO 140
      COMMON/SCALE/EGSIF,EGSIL                                          REJO 150
      DIMENSION IRESTR(3,50),IERROR(5,50),IRES(3),JNUM(50)              REJO 160
      DATA KZ,KB/1H0,1H /                                               REJO 170
C                                                                       REJO 180
C     INITIALIZE VALUES TO BE USED IN SUBROUTINE                        REJO 190
C                                                                       REJO 200
      IC=0                                                              REJO 210
      IIUNIT=IUNITS                                                     REJO 220
      IIERR=IERR                                                        REJO 230
      NDFJ=0                                                            REJO 240
      JJ=0                                                              REJO 250
      JBIG=0                                                            REJO 260
      XEGSIL = 1.E0                                                     REJO 270
      DO 30 I=1,NJD                                                     REJO 280
```

```
      JNUM(I)=99999
      X(I)=99999.
      Y(I)=99999.
      DO 10 J=1,3
10    IRESTR(J,I)=99999
      DO 20 J=1,5
20    IERROR(J,I)=0
30    CONTINUE
C
C     READ BLOCK 2 HEADING (REQUIRED)
C
      READ (NCRD,40) DHEAD
40    FORMAT (20A4)
C
C     READ BLOCK 2, CARD 2 (AT LEAST TWO REQUIRED) AND TEST DATA.
C
50    FORMAT (I5,5X,2E10.0,2X,3A1)
60    READ (NCRD,50) L,A,B,(IRES(K),K=1,3)
      IRT=1
      IF ((IRES(1).EQ.KZ).OR.(IRES(1).EQ.KB)) IRT=0
      IRES(1)=IRT
      IRT=1
      IF ((IRES(2).EQ.KZ).OR.(IRES(2).EQ.KB)) IRT=0
      IRES(2)=IRT
      IRT=1
      IF ((IRES(3).EQ.KZ).OR.(IRES(3).EQ.KB)) IRT=0
      IRES(3)=IRT
C
C     PERFORM CHECKS ON DATA AND STORE.
C
```

```
C
C     CHECK FOR LAST CARD
      IF (L.EQ.0) GO TO 130
C
C     CHECK FOR JOINT NUMBER LARGER THAN ALLOWED BY PROGRAM
      IF (L.LE.NJD) GO TO 70
C
C     IF INPUT JOINT NUMBER TOO LARGE, SET JOINT NUMBER EQUAL TO MAXIMUM
C     ALLOWABLE AND SET ERROR FLAG
      L=NJD
      IERROR(1,L)=1
      IERR=IERR+1
      GO TO 120
C
C     CHECK FOR REPEATED JOINT NUMBERS WITH CONFLICTING DATA
70    IF (JNUM(L).EQ.99999) GO TO 80
      IF (JNUM(L).EQ.L.AND.X(L).EQ.A.AND.Y(L).EQ.B.AND.IRESTR(1,L).EQ.IR
     1ES(1).AND.IRESTR(2,L).EQ.IRES(2).AND.IRESTR(3,L).EQ.IRES(3)) GO TO
     2 60
      IERROR(3,L)=1
      IERR=IERR+1
      GO TO 60
C
C     DETERMINE LARGEST JOINT NUMBER INPUT
80    IF (L.GT.JBIG) JBIG=L
C
C     INCREMENT JOINT COUNTER
      JJ=JJ+1
C
C     STORE JOINT NUMBER
```

```
      JNUM(L)=L                                                         REJO 890
C                                                                       REJO 900
C     STORE JOINT COORDINATES.                                          REJO 910
      X(L) = A                                                          REJO 920
      Y(L) = B                                                          REJO 930
C                                                                       REJO 940
C     CHECK FOR JOINTS WITH NO COORDINATES.                             REJO 950
      IF ((A.EQ.0.E0).AND.(B.EQ.0.E0)) IC=IC+1                          REJO 960
      IF (IC.LE.1) GO TO 100                                            REJO 970
      IC=IC-1                                                           REJO 980
      IERROR(2,L)=1                                                     REJO 990
      IERR=IERR+1                                                       REJO1000
C                                                                       REJO1010
C     STORE JOINT RESTRAINTS                                            REJO1020
100   DO 110 I=1,3                                                      REJO1030
110   IRESTR(I,L)=IRES(I)                                               REJO1040
120   CONTINUE                                                          REJO1050
C                                                                       REJO1060
C     RETURN TO READ NEXT JOINT COORDINATE DATA CARD                    REJO1070
      GO TO 60                                                          REJO1080
C                                                                       REJO1090
C     MAKE TOTAL SET CHECKS AND STORE DATA.                             REJO1100
C                                                                       REJO1110
C                                                                       REJO1120
C     NUMBER OF JOINTS EQUALS LARGEST JOINT NUMBER INPUT                REJO1130
130   NJ=JBIG                                                           REJO1140
C                                                                       REJO1150
C     CHECK THAT AT LEAST TWO JOINTS HAVE BEEN INPUT                    REJO1160
      IF (NJ.GE.2) GO TO 150                                            REJO1170
      IERR=IERR+1                                                       REJO1180
```

```
      CALL PAGE                                                         REJ01190
      PRINT 140                                                         REJ01200
140   FORMAT (1H ,77H*** LESS THAN TWO JOINTS INPUT TO THE JOINT COORDINREJ01210
     1ATE DATA BLOCK (REJO). ***)                                       REJ01220
      GO TO 490                                                         REJ01230
C                                                                       REJ01240
C     CHECK THAT ALL JOINT NUMBERS UP TO LARGEST JOINT NUMBER INPUT HAVEREJ01250
150   DO 200 I=1,NJ                                                     REJ01260
      IF (JNUM(I).NE.99999) GO TO 160                                   REJ01270
      IERROR(4,I)=1                                                     REJ01280
      IERR=IERR+1                                                       REJ01290
      GO TO 200                                                         REJ01300
C                                                                       REJ01310
C     COUNT DEGREES OF FREEDOM AND CHECK THAT THIS NUMBER DOES NOT EXCEEREJ01320
C     MAXIMUM ALLOWABLE                                                 REJ01330
160   DO 190 K=1,3                                                      REJ01340
      IF (IRESTR(K,I).NE.0) GO TO 180                                   REJ01350
      NDFJ=NDFJ+1                                                       REJ01360
      IF (NDFJ.LE.NDFD) GO TO 170                                       REJ01370
      NDFJ=NDFD                                                         REJ01380
      IERROR(5,I)=1                                                     REJ01390
      IERR=IERR+1                                                       REJ01400
C                                                                       REJ01410
C     STORE DEGREE OF FREEDOM NUMBER IN RESTRAINT CODE STORAGE LOCATION REJ01420
170   IRESTR(K,I)=NDFJ                                                  REJ01430
C                                                                       REJ01440
C     STORE DEGREE OF FREEDOM LOCATION                                  REJ01450
      IDFI(NDFJ)=I                                                      REJ01460
      IDFII(NDFJ)=K                                                     REJ01470
      GO TO 190                                                         REJ01480
```

```
C                                                                       REJ01490
C     OUTPUT BLOCK DATA IN PRINTED FORM.                                REJ01500
C                                                                       REJ01510
180   IF (IRESTR(K,I).EQ.1) IRESTR(K,I)=0                               REJ01520
190   CONTINUE                                                          REJ01530
200   CONTINUE                                                          REJ01540
      NDF=NDFJ                                                          REJ01550
210   CONTINUE                                                          REJ01560
      CALL PAGE                                                         REJ01570
220   FORMAT (1H ,20A4,//)                                              REJ01580
      WRITE (NPRT,220) DHEAD                                            REJ01590
C                                                                       REJ01600
C     OUTPUT JOINT COORDINATE DATA BLOCK HEADING                        REJ01610
230   FORMAT (1H ,32X,32HJOINT COORDINATES AND RESTRAINTS//1H ,9HJOINT NREJ01620
     1O.,5X,12HX-COORDINATE,5X,12HY-COORDINATE,5X,14HX-DISPLACEMENT,5X,1REJ01630
     24HY-DISPLACEMENT,6X,10HZ-ROTATION)                                REJ01640
      WRITE (NPRT,230)                                                  REJ01650
      LINE=LINE+3                                                       REJ01660
C                                                                       REJ01670
C     CHECK FOR OUTPUT UNITS (ENGLISH OR METRIC)                        REJ01680
      IF(IIUNIT.LE.1) WRITE(NPRT,260)                                   REJ01690
      IF(IIUNIT.GE.2) WRITE(NPRT,240)                                   REJ01700
240   FORMAT (18X,6HMETERS,11X,6HMETERS,/)                              REJ01710
260   FORMAT (18X,3HIN.,14X,3HIN.,/)                                    REJ01720
      LINE=LINE+2                                                       REJ01730
      DO 460 J=1,NJ                                                     REJ01740
      ITAG = 1                                                          REJ01750
C                                                                       REJ01760
C     CONVERT UNITS IF NECESSARY.                                       REJ01770
      XJ = X(J)*XEGSIL                                                  REJ01780
```

```
      YJ = Y(J)*XEGSIL                                                  REJ01790
C                                                                       REJ01800
C     OUTPUT METRIC UNIT DATA                                           REJ01810
  290 IF(IIUNIT.EQ.0.OR.IIUNIT.EQ.1) GO TO 340                          REJ01820
      WRITE(NPRT,350) JNUM(J),XJ,YJ,(IRESTR(K,J),K=1,3)                 REJ01830
C                                                                       REJ01840
C     REPRINT PAGE AND TABLE HEADINGS, IF NEEDED.                       REJ01850
300   LINE=LINE+1                                                       REJ01860
      IF (LINE.LE.NL) GO TO 330                                         REJ01870
      CALL PAGE                                                         REJ01880
      WRITE (NPRT,230)                                                  REJ01890
      IF(IIUNIT.EQ.0.OR.IIUNIT.EQ.3) GO TO 310                          REJ01900
      WRITE (NPRT,240)                                                  REJ01910
      GO TO 320                                                         REJ01920
310   WRITE (NPRT,260)                                                  REJ01930
320   LINE=LINE+5                                                       REJ01940
330   GO TO (360,380,400,420,440,460), ITAG                             REJ01950
C                                                                       REJ01960
C     OUTPUT ENGLISH UNIT DATA                                          REJ01970
340   WRITE(NPRT,350) JNUM(J),XJ,YJ,(IRESTR(K,J),K=1,3)                 REJ01980
350   FORMAT (1H ,I5,9X,0PE12.5,5X,0PE12.5,7X,I5,14X,I5,13X,I5)         REJ01990
      GO TO 300                                                         REJ02000
C                                                                       REJ02010
C     OUTPUT ERROR MESSAGES                                             REJ02020
360   IF (IERROR(2,J).EQ.0) GO TO 380                                   REJ02030
370   FORMAT (1H ,9H***JOINT ,I5,91H IS UNACCEPTABLE SINCE A JOINT IS ALREJ02040
     1READY GIVEN AT THE ORIGIN OF THE COORDINATES (REJO).***)          REJ02050
      PRINT 370, JNUM(J)                                                REJ02060
      ITAG=2                                                            REJ02070
      GO TO 300                                                         REJ02080
```

```
380   IF (IERROR(3,J).EQ.0) GO TO 400                                REJO2090
390   FORMAT (1H ,46H*** TWO DIFFERENT SETS OF DATA INPUT FOR JOINT,I5,1REJO2100
     13H  (REJO). ***)                                               REJO2110
      PRINT 390, JNJM(J)                                             REJO2120
      ITAG=3                                                         REJO2130
      GO TO 300                                                      REJO2140
400   IF (IERROR(4,J).EQ.0) GO TO 420                                REJO2150
      PRINT 410, J                                                   REJO2160
410   FORMAT (1H ,16H*** JOINT NUMBER,I5,22H NOT INPUT (REJO). ***)  REJO2170
      ITAG=4                                                         REJO2180
      GO TO 300                                                      REJO2190
420   IF (IERROR(5,J).EQ.0) GO TO 440                                REJO2200
430   FORMAT (1H ,37H*** NO. OF DEGREES OF FREEDOM EXCEEDS,I5,65H,LARGESREJO2210
     1T NO. OF DEGREES OF FREEDOM ALLOWED BY PROGRAM (REJO). ***)    REJO2220
      PRINT 430, NDFD                                                REJO2230
      ITAG=5                                                         REJO2240
      GO TO 300                                                      REJO2250
440   IF (IERROR(1,J).EQ.0) GO TO 460                                REJO2260
450   FORMAT (1H ,36H*** JOINT NUMBER INPUT WHICH EXCEEDS,I5,53H, LARGESREJO2270
     1T JOINT NUMBER ALLOWED BY PROGRAM (REJO). ***)                 REJO2280
      PRINT 450, NJD                                                 REJO2290
      ITAG=6                                                         REJO2300
      GO TO 300                                                      REJO2310
460   CONTINUE                                                       REJO2320
C                                                                    REJO2330
C     IF INPUT UNITS MATCH OUTPUT UNITS, FINISHED                    REJO2340
      IF (IIUNIT.EQ.0.OR.IIUNIT.EQ.2) GO TO 490                      REJO2350
C                                                                    REJO2360
C     IF INPUT ERRORS ENCOUNTERED, FINISHED                          REJO2370
      IF (IERR.GT.IIERR) GO TO 490                                   REJO2380
```

```
C                                                                       REJ02390
C     OUTPUT JOINT DATA IN DIFFERENT DIMENSIONAL UNITS, IF NECESSARY.   REJ02400
C                                                                       REJ02410
C     IF INPUT AND OUTPUT UNITS SPECIFIED DO NOT MATCH, OUTPUT OTHER SETREJ02420
      XEGSIL = EGSIL                                                    REJ02430
      IF(IIUNIT.EQ.1) IIUNIT = 2                                        REJ02440
      IF(IIUNIT.E(.3) IIUNIT = 0                                        REJ02450
      GO TO 210                                                         REJ02460
490   RETURN                                                            REJ02470
      END                                                               REJ02480
```

```
CSECT    0  10
      SUBROUTINE SECT                                                   SECT   0
C                                                                       SECT  10
C THIS SUBROUTINE LOCATES INTEGRATION SECTIONS, PREPARES FIBER DATA,    SECT  20
C AND SMEARS OUT LATERAL AND LONGITUDINAL REINFORCEMENT.                SECT  30
C                                                                       SECT  40
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),       SECT  50
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),SECT  60
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)                      SECT  70
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,SECT  80
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,SECT  90
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, SECT 100
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  SECT 110
     4    NTIMES,NVEL,IINITD                                            SECT 120
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),SECT 130
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),    SECT 140
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),SECT 150
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                SECT 160
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45), SECT 170
     5    YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)              SECT 180
C                                                                       SECT 190
      GAUSS2 = 0.57735026918963E0                                       SECT 200
      GAUSS3 = 0.77459666924148E0                                       SECT 210
C                                                                       SECT 220
C BEGIN DO LOOP OVER MEMBER NUMBER.                                     SECT 230
      DO 100 M=1,NM                                                     SECT 240
C                                                                       SECT 250
C DETERMINE X-DISTANCES FROM P-END OF MEMBER TO INTEGRATION SECTION.    SECT 260
      XLH=0.5E0*XL(M)                                                   SECT 270
      XLA=GAUSS3*XLH                                                    SECT 280
```

```
      XPI(1,M)=XLH-XLA                                                  SECT 290
      XPI(2,M)=XLH                                                      SECT 300
      XPI(3,M)=XLH+XLA                                                  SECT 310
C                                                                       SECT 320
C CALCULATE Y-DISTANCES FROM REFERENCE AXIS FOR EACH FIBER.             SECT 330
      IF (MTYPE(M).EQ.4) GO TO 80                                       SECT 340
      IF (NTIES(M).NE.0) GO TO 20                                       SECT 350
      COVR=HMEM(M)-D(M)                                                 SECT 360
      DPP(M)=HMEM(M)-2.E0*COVR                                          SECT 370
      GO TO 30                                                          SECT 380
20    COVR=(HMEM(M)-DPP(M))/2.E0                                        SECT 390
30    YFIBR(1,M)=HTOP(M)                                                SECT 400
      YFIBR(11,M)=HTOP(M)-HMEM(M)                                       SECT 410
      YFIBR(2,M)=HTOP(M)-COVR                                           SECT 420
      DINC=DPP(M)/8.E0                                                  SECT 430
      DO 40 I=3,10                                                      SECT 440
40    YFIBR(I,M)=YFIBR(I-1,M)-DINC                                      SECT 450
      YFG=0.5E0*COVR*GAUSS2                                             SECT 460
      YFH=HTOP(M)-0.5E0*COVR                                            SECT 470
      YWG = (0.5E0*HMEM(M)-COVR)*GAUSS3                                 SECT 480
      YWH=HTOP(M)-0.5E0*HMEM(M)                                         SECT 490
      YFB=HTOP(M)-HMEM(M)+0.5E0*COVR                                    SECT 500
C                                                                       SECT 510
C SMEAR OUT LATERAL REINFORCEMENT FOR R/C  MEMBER.                      SECT 520
      NT=NTIES(M)                                                       SECT 530
      IF (NT.EQ.0) GO TO 60                                             SECT 540
      NSPACE=0                                                          SECT 550
      PDPS=0.E0                                                         SECT 560
      PPFS=0.E0                                                         SECT 570
      DO 50 J=1,NT                                                      SECT 580
```

```
      PDPS=PDPS+PDP(J,M)
      PPFS=PPFS+PDF(J,M)
50    NSPACE=NSPACE+NSPAC(J,M)
      STIES(7,M)=XL(M)/FLOAT(NSPACE)
      PDP(7,M)=PDPS/FLOAT(NT)
      PDF(7,M)=PPFS/FLOAT(NT)
C
C SMEAR OUT LONGITUDINAL REINFORCEMENT FOR R/C  MEMBER.
60    NG=NGRP(M)
      IF (NG.EQ.0) GO TO 95
      DO 70 K=1,NG
70    AGRP(K,M)=AGRP(K,M)*EFFL(K,M)/XL(M)
      GO TO 95
C
C WIDE FLANGE MEMBER.
80    YFIBR(1,M)=HTWF(M)
      YFIBR(2,M)=HTWF(M)-TFWF(M)
      YFIBR(11,M)=HTWF(M)-DWF(M)
      YFIBR(10,M)=YFIBR(11,M)+TFWF(M)
      DINC=(YFIBR(2,M)-YFIBR(10,M))/8.E0
      DO 90 J=3,9
90    YFIBR(J,M)=YFIBR(J-1,M)-DINC
      YFG=0.5E0*GAUSS2*TFWF(M)
      YFH=HTWF(M)-0.5E0*TFWF(M)
      YWG=(0.5E0*DWF(M)-TFWF(M))*GAUSS3
      YWH=HTWF(M)-0.5E0*DWF(M)
      YFB=HTWF(M)-DWF(M)+0.5E0*TFWF(M)
95    YGP(1,M)=YFH+YFG
      YGP(2,M)=YFH-YFG
      YGP(3,M)=YWH+YWG
```

```
      YGP(4,M)=YWH                                                      SECT 890
       YGP(5,M)=YWH-YWG                                                 SECT 900
       YGP(6,M)=YFB+YFG                                                 SECT 910
      YGP(7,M)=YFB-YFG                                                  SECT 920
100    CONTINUE                                                         SECT 930
      RETURN                                                            SECT 940
C                                                                       SECT 950
C ***** GLOSSARY FOR SECT *****                                         SECT 960
C                                                                       SECT 970
C COVR      = CONCRETE COVER FOR TOP AND BOTTOM REINFORCEMENT.          SECT 980
C DINC      = DEPTH INCREMENT FOR LOCATING CONCRETE FIBERS.             SECT 990
C NG        = NUMBER OF LONGITUDINAL REINFORCEMENT GROUPS IN MEMBER.    SECT1000
C NSPACE    = TOTAL NUMBER OF STIRRUP SPACINGS IN MEMBER.               SECT1010
C NT        = NUMBER OF LATERAL REINFORCEMENT GROUPS IN MEMBER.         SECT1020
C XPI       = X-DISTANCE OF INTEGRATION SECTION FROM P-END OF MEMBER.   SECT1030
C YFIBR     = FIBER Y-DISTANCE FROM REFERENCE AXIS.                     SECT1040
C                                                                       SECT1050
      END                                                               SECT1060
```

```
CSEEK    0  10
      SUBROUTINE SEEK (DEFORM,ENGY)
C
C     THIS SUBROUTINE OBTAINS THE DISPLACEMENT SOLUTION FOR THE CURRENT
C     TIME STEP.A FUNCTION MINIMIZATION PROCEDURE IS EMPLOYED.
C
      COMMON DATA(10000),KDATA(500)
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20),
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,
     4    NTIMES,NVEL,IINITD
      COMMON/SEEKBK/DEFOR(90),STPSIZ(90),GRAD(90),GRADI(90),DELTAG(90),
     1    DIRECT(90),DIAG(90),STEP(4),DSTEP(4),FVAL(4),VALUES(7),
     2    DISACC,SSIZE,FUNACC,FUNMIN,CRITL,CRITU,NLIN
C
      INTEGER HEAD,DHEAD
      DIMENSION DEFORM(90),CURV(4095)
C
      EQUIVALENCE (DATA(1),CURV(1))
C
C     INITIALIZE
C
      IF (IINITD.GE.0) GO TO 40
      IINITD = 0
```

```
      READ (NCRD,10) (DEFORM(I),I=1,NDF)                                SEEK 290
10    FORMAT (8E10.0)                                                   SEEK 300
      WRITE (NPRT,20)                                                   SEEK 310
20    FORMAT (47H THE FOLLOWING INITIAL GUESS (SCALED) WAS USED-)       SEEK 320
      WRITE (NPRT,30) (DEFORM(I),I=1,NDF)                               SEEK 330
30    FORMAT (1H ,(25X,1P5E17.9))                                       SEEK 340
C                                                                       SEEK 350
C     SET CURVATURES EQUAL TO THE IDENTITY MATRIX IF THIS IS THE FIRST USEEK 360
C     OF MINMIZ FOR THE CURRENT PROBLEM (ICURV=1). OTHERWISE, USE THE   SEEK 370
C     PREVIOUS CURVATURE MATRIX (ICURV=0).                              SEEK 380
C                                                                       SEEK 390
40    IF (ICURV.EQ.0) GO TO 70                                          SEEK 400
      ICURV = 0                                                         SEEK 410
      DISACC=ABS(DISACC)                                                SEEK 420
      SSIZE=ABS(SSIZE)                                                  SEEK 430
      K=NDF*(NDF+1)/2                                                   SEEK 440
      DO 50 I=1,K                                                       SEEK 450
50    CURV(I)=0.E0                                                      SEEK 460
      DO 60 I=1,NDF                                                     SEEK 470
      K=NDF*(I-1)+I-I*(I-1)/2                                           SEEK 480
      CURV(K)=1.E0                                                      SEEK 490
      DIAG(I)=1.E0                                                      SEEK 500
60    STPSIZ(I)=SSIZE                                                   SEEK 510
70    KLIN = 1                                                          SEEK 520
      KOUNT=0                                                           SEEK 530
      NCOUNT=0                                                          SEEK 540
      MINIM=1                                                           SEEK 550
      IDENT=1                                                           SEEK 560
      STEP(1)=1.E0                                                      SEEK 570
      EMAX=0.E0                                                         SEEK 580
```

```
C                                                                               SEEK 590
C     INITIAL FUNCTION VALUE                                                    SEEK 600
C                                                                               SEEK 610
      CALL POTE (DEFORM,ENGY)                                                   SEEK 620
      KOUNT=KOUNT+1                                                             SEEK 630
      WRITE (NPRT,80)                                                           SEEK 640
80    FORMAT (25H0MINIMIZATION PERFORMANCE///)                                  SEEK 650
      IF (IPRINT.LE.1) GO TO 120                                                SEEK 660
      WRITE (NPRT,100)                                                          SEEK 670
100   FORMAT (14H INITIAL POINT)                                                SEEK 680
      WRITE (NPRT,110) ENGY,(DEFORM(I),I=1,NDF)                                 SEEK 690
110   FORMAT (16H  FUNCTION VALUE,9X,9HVARIABLES/3X,0PE17.9,5X,5E17.9/(2SEEK 700
     15X,5E17.9))                                                               SEEK 710
C                                                                               SEEK 720
C     INITIAL GRADIENT                                                          SEEK 730
C                                                                               SEEK 740
120   IF(IPRINT.GT.0) WRITE(NPRT,130) KLIN                                      SEEK 750
130   FORMAT (/20H0LINEAR MINIMIZATION,I4)                                      SEEK 760
      ILOOP=1                                                                   SEEK 770
140   GRADM=0.E0                                                                SEEK 780
      IF (IPRINT.GT.2) WRITE (NPRT,150)                                         SEEK 790
150   FORMAT (10H  GRADIENT,15X,35HSTEPSIZE USED IN COMPUTING GRADIENT) SEEK 800
      DO 170 I=1,NDF                                                            SEEK 810
      CALL DELT (DEFORM,STPSIZ(I),I,ENGG,0)                                     SEEK 820
      NCOUNT=NCOUNT+1                                                           SEEK 830
      GRADM=GRADM+ENGG*ENGG                                                     SEEK 840
      IF (IPRINT.GT.2) WRITE (NPRT,160) ENGG,STPSIZ(I)                          SEEK 850
160   FORMAT (3X,0PE17.9,5X,E17.9)                                              SEEK 860
170   GRAD(I)=ENGG                                                              SEEK 870
      IF (GRADM.NE.0.E0) GO TO 220                                              SEEK 880
```

```
      ILOOP=ILOOP+1                                                     SEEK 890
      IF (ILOOP.GT.5) GO TO 190                                         SEEK 900
      DO 180 I=1,NDF                                                    SEEK 910
180   STPSIZ(I)=STPSIZ(I)*8.E0                                          SEEK 920
      GO TO 140                                                         SEEK 930
190   WRITE (NPRT,200)                                                  SEEK 940
200   FORMAT (32H0ZERO INITIAL GRADIENT IN MINMIZ)                      SEEK 950
      WRITE (NPRT,210)                                                  SEEK 960
210   FORMAT (73H *** ABNORMAL COMPLETION OF MINIMIZATION, ANALYSIS TERMSEEK 970
     1INATED (SEEK). ***)                                               SEEK 980
      IERR=1                                                            SEEK 990
      GO TO 940                                                         SEEK1000
220   IF (IPRINT.EQ.2) WRITE (NPRT,230) (GRAD(I), I=1,NDF)              SEEK1010
230   FORMAT (12H  GRADIENTS,,13X,1P5E17.9/(25X,5E17.9))                SEEK1020
      IF (IPRINT.GT.1) WRITE (NPRT,240) (DIAG(I),I=1,NDF)               SEEK1030
240   FORMAT (12H  CURVATURES,13X,1P5E17.9/(25X,5E17.9))                SEEK1040
C                                                                       SEEK1050
C     SET UP FOR A MINIMIZATION ALONG A LINE                            SEEK1060
C                                                                       SEEK1070
250   DX=0.E0                                                           SEEK1080
      EPVAL=1.E0                                                        SEEK1090
      EQVAL=1.E0                                                        SEEK1100
      K=0                                                               SEEK1110
      DO 290 I=1,NDF                                                    SEEK1120
      DIRECT(I)=0.E0                                                    SEEK1130
      DO 260 J=I,NDF                                                    SEEK1140
      K=K+1                                                             SEEK1150
260   DIRECT(I)=DIRECT(I)-CURV(K)*GRAD(J)                               SEEK1160
      IF (I.EQ.1) GO TO 280                                             SEEK1170
      J1=I-1                                                            SEEK1180
```

```
      DO 270 J=1,J1                                                     SEEK1190
      L=NDF*(J-1)+I-J*(J-1)/2                                           SEEK1200
  270 DIRECT(I)=DIRECT(I)-CURV(L)*GRAD(J)                               SEEK1210
  280 IF (DIRECT(I).EQ.0.E0) GO TO 290                                  SEEK1220
      EPVAL = AMIN1(EPVAL,ABS(DISACC/DIRECT(I)))                        SEEK1230
      EQVAL = AMIN1(EQVAL,(EPS/10.E0)*ABS(DEFORM(I)/DIRECT(I)))         SEEK1240
      DX=DX+GRAD(I)*DIRECT(I)                                           SEEK1250
  290 CONTINUE                                                          SEEK1260
      EPVAL=.05E0*EPVAL                                                 SEEK1270
C                                                                       SEEK1280
C       IF DIRECTION IS NOT DOWNHILL, SET CURVATURES EQUAL TO IDENTITY. SEEK1290
C                                                                       SEEK1300
      IF (DX.LT.0.E0) GO TO 340                                         SEEK1310
      IREC=IREC+1                                                       SEEK1320
      PRINT 300                                                         SEEK1330
  300 FORMAT (93H * THE PREDICTED DIRECTION IS NOT DOWNHILL, CURVATURES SEEK1340
     1 SET EQUAL TO IDENTITY MATRIX (SEEK).* )                          SEEK1350
      DX = 0.E0                                                         SEEK1360
      EPVAL=1.E0                                                        SEEK1370
      EQVAL=1.E0                                                        SEEK1380
      K=NDF*(NDF+1)/2                                                   SEEK1390
      DO 310 I=1,K                                                      SEEK1400
  310 CURV(I)=0.E0                                                      SEEK1410
      DO 320 I=1,NDF                                                    SEEK1420
      K=NDF*(I-1)+I-I*(I-1)/2                                           SEEK1430
      CURV(K)=1.E0                                                      SEEK1440
      DIAG(I)=1.E0                                                      SEEK1450
      DIRECT(I)=-GRAD(I)                                                SEEK1460
      IF (DIRECT(I).EQ.0.E0) GO TO 320                                  SEEK1470
      EPVAL = AMIN1(EPVAL,ABS(DISACC/DIRECT(I)))                        SEEK1480
```

```
      EQVAL = AMIN1(EQVAL,(EPS/10.E0)*ABS(DEFORM(I)/DIRECT(I)))          SEEK1490
      DX=DX+GRAD(I)*DIRECT(I)                                            SEEK1500
  320 CONTINUE                                                           SEEK1510
      EPVAL=.05E0*EPVAL                                                  SEEK1520
      IF (IPRINT.GT.1) WRITE (NPRT,240) (DIAG(I),I=1,NDF)                SEEK1530
      IF (IPRINT.GT.0) WRITE (NPRT,330) (DIRECT(I),I=1,NDF)              SEEK1540
  330 FORMAT (12H  DIRECTIONS,13X,1P5E17.9/(25X,5E17.9))                 SEEK1550
C                                                                        SEEK1560
C     MAKE FIRST STEP ALONG LINE                                         SEEK1570
C                                                                        SEEK1580
  340 IF (ENGY.LE.FUNMIN) GO TO 350                                      SEEK1590
      STEP(2)=2.E0*(FUNMIN-ENGY)/DX                                      SEEK1600
      IF (STEP(2).LE.1.E0) GO TO 360                                     SEEK1610
  350 STEP(2)=1.E0                                                       SEEK1620
  360 STEP(2)=AMAX1(STEP(2),EQVAL)                                       SEEK1630
      FVAL(1)=ENGY                                                       SEEK1640
      STEP(1)=0.E0                                                       SEEK1650
      KKK=0                                                              SEEK1660
      IF (IPRINT.GT.1) WRITE (NPRT,370)                                  SEEK1670
  370 FORMAT (16H  FUNCTION VALUE,9X,27HFRACTION OF DIRECTION TAKEN)     SEEK1680
      ILOOP=1                                                            SEEK1690
  380 DO 390 I=1,NDF                                                     SEEK1700
  390 DEFOR(I)=DEFORM(I)+STEP(2)*DIRECT(I)                               SEEK1710
      CALL POTE (DEFOR,FVAL(2))                                          SEEK1720
      KOUNT=KOUNT+1                                                      SEEK1730
      IF (IPRINT.GT.1) WRITE (NPRT,160) FVAL(2),STEP(2)                  SEEK1740
C                                                                        SEEK1750
C     TEST FOR CHANGE IN FUNCTION VALUE                                  SEEK1760
C                                                                        SEEK1770
      IF (FVAL(2).NE.FVAL(1)) GO TO 430                                  SEEK1780
```

```
      ILOOP=ILOOP+1                                                     SEEK1790
      IF (ILOOP.LE.5) GO TO 420                                         SEEK1800
      WRITE (NPRT,400)                                                  SEEK1810
400   FORMAT (40H NO CHANGE IN FUNCTION VALUE ALONG LINE.)              SEEK1820
      WRITE (NPRT,410)                                                  SEEK1830
410   FORMAT (35H NORMAL COMPLETION OF MINIMIZATION.)                   SEEK1840
      GO TO 920                                                         SEEK1850
420   STEP(2)=8.E0*STEP(2)                                              SEEK1860
      GO TO 380                                                         SEEK1870
C                                                                       SEEK1880
C     MAKE SECOND STEP ALONG LINE BY PARABOLIC INTERPOLATION.           SEEK1890
C                                                                       SEEK1900
430   STEP2=0.5E0*DX*STEP(2)**2/(DX*STEP(2)+(FVAL(1)-FVAL(2)))          SEEK1910
      IF (STEP2.LE.0.E0) STEP2=2.E0*STEP(2)                             SEEK1920
      IF (FVAL(2).LT.FVAL(1)) GO TO 450                                 SEEK1930
      STEP(2)=STEP2                                                     SEEK1940
      KKK=KKK+1                                                         SEEK1950
      IF (KKK.LT.2) GO TO 380                                           SEEK1960
      FVAL(3)=FVAL(2)                                                   SEEK1970
      FVAL(2)=ENGY                                                      SEEK1980
      STEP(3)=STEP(2)                                                   SEEK1990
      STEP(2)=0.E0                                                      SEEK2000
      STEP(1)=-STEP(3)                                                  SEEK2010
      DO 440 I=1,NDF                                                    SEEK2020
440   DEFOR(I)=DEFORM(I)+STEP(1)*DIRECT(I)                              SEEK2030
      CALL POTE (DEFOR,FVAL(1))                                         SEEK2040
      KOUNT=KOUNT+1                                                     SEEK2050
      IF (IPRINT.GT.1) WRITE (NPRT,160) FVAL(1),STEP(1)                 SEEK2060
      GO TO 480                                                         SEEK2070
450   MINIM=2                                                           SEEK2080
```

```
      IF (STEP2.GT.4.E0*STEP(2)) STEP2=4.E0*STEP(2)                     SEEK2090
      IF (ABS(STEP(2)-STEP2).LT.EPVAL) STEP2=STEP(2)+1.1E0*EPVAL        SEEK2100
      IF (ABS(STEP(2)-STEP2).LT..03E0*ABS(STEP(2)))STEP2=1.1E0*STEP(2)  SEEK2110
      DO 460 I=1,NDF                                                    SEEK2120
  460 DEFOR(I)=DEFORM(I)+STEP2*DIRECT(I)                                SEEK2130
      IF (STEP2.GT.STEP(2)) GO TO 470                                   SEEK2140
      STEP(3)=STEP(2)                                                   SEEK2150
      STEP(2)=STEP2                                                     SEEK2160
      FVAL(3)=FVAL(2)                                                   SEEK2170
      CALL POTE (DEFOR,FVAL(2))                                         SEEK2180
      KOUNT=KOUNT+1                                                     SEEK2190
      IF (IPRINT.GT.1) WRITE (NPRT,160) FVAL(2),STEP(2)                 SEEK2200
      GO TO 480                                                         SEEK2210
  470 STEP(3)=STEP2                                                     SEEK2220
      CALL POTE (DEFOR,FVAL(3))                                         SEEK2230
      KOUNT=KOUNT+1                                                     SEEK2240
      IF (IPRINT.GT.1) WRITE (NPRT,160) FVAL(3),STEP(3)                 SEEK2250
C                                                                       SEEK2260
C     DETERMINE LOCATION OF MINIMUM ALONG LINE                          SEEK2270
C                                                                       SEEK2280
  480 INIT=0                                                            SEEK2290
      CALL FITS (STEP,FVAL,DSTEP,AVAL,INIT)                             SEEK2300
  490 MINIM=1                                                           SEEK2310
      DO 500 I=2,3                                                      SEEK2320
  500 IF (FVAL(I).LT.FVAL(MINIM)) MINIM=I                               SEEK2330
      IE=2.E0+SIGN(1.E0,DSTEP(2))                                       SEEK2340
      IF (AVAL.EQ.0.E0) IE=2.E0+SIGN(1.E0,FVAL(1)-FVAL(2))              SEEK2350
      IF (AVAL.LT.0.E0) IE=4-IE                                         SEEK2360
      IF (AVAL.LE.0.E0.OR.ABS(DSTEP(2)).GT.ABS(4.E0*DSTEP(IE))) DSTEP(  SEEK2370
     12)=4.E0*DSTEP(IE)                                                 SEEK2380
```

```
      STEP3=STEP(2)+DSTEP(2)                                            SEEK2390
      IF (ABS(STEP3-STEP(MINIM)).LT.EPVAL) GO TO 580                    SEEK2400
      IF (ABS(STEP3-STEP(MINIM)).LT..03E0*ABS(STEP(MINIM))) GO TO 580   SEEK2410
      IF (DSTEP(IE).LT.DSTEP(2)) IE=IE+1                                SEEK2420
      IF (IE.EQ.4) GO TO 520                                            SEEK2430
      DO 510 LL=IE,3                                                    SEEK2440
      L=3-LL+IE                                                         SEEK2450
      STEP(L+1)=STEP(L)                                                 SEEK2460
  510 FVAL(L+1)=FVAL(L)                                                 SEEK2470
  520 STEP(IE)=STEP3                                                    SEEK2480
      DO 530 I=1,NDF                                                    SEEK2490
  530 DEFOR(I)=DEFORM(I)+STEP3*DIRECT(I)                                SEEK2500
      CALL POTE (DEFOR,FVAL(IE))                                        SEEK2510
      KOUNT=KOUNT+1                                                     SEEK2520
      IF (IPRINT.GT.1) WRITE (NPRT,160) FVAL(IE),STEP(IE)               SEEK2530
      IF (IE.EQ.1) GO TO 480                                            SEEK2540
      KKK=1                                                             SEEK2550
      IF (IE.EQ.4) GO TO 560                                            SEEK2560
      IF (FVAL(1).GT.FVAL(4)) GO TO 540                                 SEEK2570
      INIT=0                                                            SEEK2580
      CALL FITS (STEP,FVAL,DSTEP,AVAL,INIT)                             SEEK2590
      IF (STEP(2)+DSTEP(2).LT.STEP(4).AND.AVAL.GT.0.E0) GO TO 490       SEEK2600
      GO TO 550                                                         SEEK2610
  540 KKK=2                                                             SEEK2620
      INIT=1                                                            SEEK2630
      CALL FITS (STEP,FVAL,DSTEP,AVAL,INIT)                             SEEK2640
      IF (STEP(3)+DSTEP(2).GT.STEP(1).AND.AVAL.GT.0.E0) GO TO 560       SEEK2650
  550 KKK=1                                                             SEEK2660
      IF (FVAL(2).LT.FVAL(1).AND.FVAL(2).LE.FVAL(3).OR.FVAL(2).LE.FVAL(1SEEK2670
     1).AND.FVAL(2).LT.FVAL(3)) GO TO 480                               SEEK2680
```

```
560   DO 570 I=1,3                                                      SEEK2690
      STEP(I)=STEP(I+1)                                                 SEEK2700
570   FVAL(I)=FVAL(I+1)                                                 SEEK2710
      IF (KKK.EQ.2) GO TO 490                                           SEEK2720
      GO TO 480                                                         SEEK2730
C                                                                       SEEK2740
C     END OF MINIMIZATION ALONG LINE                                    SEEK2750
C                                                                       SEEK2760
580   IF (IPRINT.EQ.1) GO TO 590                                        SEEK2770
      IF (IPRINT.LE.0) GO TO 620                                        SEEK2780
      WRITE (NPRT,600) FVAL(MINIM),STEP(MINIM)                          SEEK2790
600   FORMAT (3H  *,0PE17.9,5X,E17.9)                                   SEEK2800
590   WRITE (NPRT,610) KOUNT,NCOUNT                                     SEEK2810
610   FORMAT (28H  FUNCTION SUBROUTINE CALLS=,I5/28H  GRADIENT SUBROUTINSEEK2820
     1E CALLS=,I5)                                                      SEEK2830
C                                                                       SEEK2840
C     IF THERE WAS NO MOTION, RETURN                                    SEEK2850
C                                                                       SEEK2860
620   IF (STEP(MINIM).NE.0.E0) GO TO 640                                SEEK2870
      WRITE (NPRT,630)                                                  SEEK2880
630   FORMAT (38H NO MOTION IN THE LINEAR MINIMIZATION.)                SEEK2890
      WRITE (NPRT,410)                                                  SEEK2900
      GO TO 920                                                         SEEK2910
C                                                                       SEEK2920
C     IF THE FUNCTION VALUE HAS NOT CHANGED, RETURN                     SEEK2930
C                                                                       SEEK2940
640   IF (FVAL(MINIM).NE.ENGY) GO TO 660                                SEEK2950
      WRITE (NPRT,650)                                                  SEEK2960
650   FORMAT (57H LINEAR MINIMIZATION FAILED TO CHANGE THE FUNCTION VALUSEEK2970
     1E.)                                                               SEEK2980
```

```
      WRITE (NPRT,410)                                                  SEEK2990
      GO TO 920                                                         SEEK3000
C                                                                       SEEK3010
C     TEST FOR CONVERGENCE ANC UPDATE SOLUTION                          SEEK3020
C                                                                       SEEK3030
660   ENGY=FVAL(MINIM)                                                  SEEK3040
      ETEST=AMAX1(1.E0,ABS(STEP(MINIM)))                                SEEK3050
      EMAX=0.E0                                                         SEEK3060
      DO 670 I=1,NDF                                                    SEEK3070
      EMAX=AMAX1(EMAX,ABS(ETEST*DIRECT(I)))                             SEEK3080
      DIRECT(I)=STEP(MINIM)*DIRECT(I)                                   SEEK3090
      DEFORM(I)=DEFORM(I)+DIRECT(I)                                     SEEK3100
670   GRADI(I)=GRAD(I)                                                  SEEK3110
      IF (IPRINT.LE.0) GO TO 700                                        SEEK3120
      WRITE (NPRT,690) EMAX                                             SEEK3130
690   FORMAT (28H  MINIMIZATION ACCURACY     =,E20.10)                  SEEK3140
      WRITE (NPRT,110) ENGY,(DEFORM(I),I=1,NDF)                         SEEK3150
700   IF (EMAX.GT.DISACC) GO TO 720                                     SEEK3160
      WRITE (NPRT,710)                                                  SEEK3170
710   FORMAT (51H CONVERGENCE OF MINIMIZATION TO SPECIFIED ACCURACY.)   SEEK3180
      WRITE (NPRT,410)                                                  SEEK3190
      GO TO 920                                                         SEEK3200
C                                                                       SEEK3210
C     NOTE IF MINIMUM WAS FOUND ALONG NEGATIVE DIRECTION                SEEK3220
C                                                                       SEEK3230
720   IF (STEP(MINIM).GE.0.E0) GO TO 740                                SEEK3240
      IF (IPRINT.LE.0) GO TO 740                                        SEEK3250
      PRINT 730                                                         SEEK3260
      IREC=IREC+1                                                       SEEK3270
730   FORMAT (54H** MINIMUM LOCATED ALONG NEGATIVE DIRECTION (SEEK). **)SEEK3280
```

```
C
C     IF THERE HAVE BEEN TOO MANY LINEAR MINIMIZATIONS, RETURN
C
740   IF (KLIN.GE.NLIN) GO TO 750
      KLIN = KLIN + 1
      GO TO 770
750   WRITE (NPRT,760)
760   FORMAT (57H THE MAXIMUM NUMBER OF LINEAR MINIMIZATIONS WAS EXCEEDE
     1D.)
      WRITE (NPRT,210)
      IERR=1
      GO TO 920
C
C CALCULATE A NEW GRADIENT BASED ON STEPSIZE DETERMINED BY
C STEWART*S MODIFICATION.
C
770   IF (IPRINT.GT.0) WRITE(NPRT,130) KLIN
      IF (IPRINT.GT.2) WRITE (NPRT,150)
      DO 810 I=1,NDF
      DEFOR(I)=DEFORM(I)
      IF (ENGY.EQ.0.E0) GO TO 800
      IF (IDENT.EQ.1) GO TO 800
      IF (GRAD(I).EQ.0.E0) GO TO 800
      ETAM=ABS(EPS*GRAD(I)*DEFORM(I)/ENGY)
      IF (FUNACC.GT.ETAM) ETAM=FUNACC
      IF (GRAD(I)**2.GT.DIAG(I)*ABS(ENGY)*ETAM) GO TO 780
      STPSIZ(I)=2.E0*(ABS(ENGY)*ABS(GRAD(I))*ETAM/DIAG(I)**2)**(1.E0
     1/3.E0)
      STPSIZ(I)=STPSIZ(I)*(1.E0-ABS(GRAD(I))/(1.5E0*DIAG(I)*STPSIZ(I)
     1+2.E0*ABS(GRAD(I))))
```

```
      GO TO 790                                                         SEEK3590
780   STPSIZ(I)=2.E0*SQRT(ETAM*ABS(ENGY)/DIAG(I))                       SEEK3600
      STPSIZ(I)=STPSIZ(I)*(1.E0-DIAG(I)*STPSIZ(I)/(3.E0*DIAG(I)*STPS    SEEK3610
     1IZ(I)+4.E0*ABS(GRAD(I))))                                         SEEK3620
790   STPSIZ(I)=SIGN(STPSIZ(I),GRAD(I))                                 SEEK3630
      IF (.5E0*ABS(DIAG(I)*STPSIZ(I)/GRAD(I)).LT..01E0) GO TO 800       SEEK3640
      STPSIZ(I)=100.E0*ABS(ENGY*ETAM/GRAD(I))                           SEEK3650
      STPSIZ(I)=ABS(GRAD(I))+SQRT(GRAD(I)**2+200.E0*ABS(ENGY)*DIAG(     SEEK3660
     1I)*ETAM)                                                          SEEK3670
      STPSIZ(I)=100.E0*ABS(ENGY)*ETAM/STPSIZ(I)                         SEEK3680
800   CALL DELT (DEFORM,STPSIZ(I),I,ENGG,0)                             SEEK3690
      NCOUNT=NCOUNT+1                                                   SEEK3700
      IF (IPRINT.GT.2) WRITE (NPRT,160) ENGG,STPSIZ(I)                  SEEK3710
810   GRAD(I)=ENGG                                                      SEEK3720
      IDENT=0                                                           SEEK3730
      IF (IPRINT.EQ.2) WRITE (NPRT,230) (GRAD(I),I=1,NDF)               SEEK3740
C                                                                       SEEK3750
C     UPDATE CURVATURES                                                 SEEK3760
C                                                                       SEEK3770
      AVAL=0.E0                                                         SEEK3780
      DO 820 I=1,NDF                                                    SEEK3790
      DELTAG(I)=GRAD(I)-GRADI(I)                                        SEEK3800
820   AVAL=AVAL+DELTAG(I)*DIRECT(I)                                     SEEK3810
      IF (AVAL.EQ.0.E0) AVAL=TINY                                       SEEK3820
      AA=AVAL/STEP(MINIM)                                               SEEK3830
      C1=1.E0/AVAL-DX/AA**2                                             SEEK3840
      C2=2.E0/AA                                                        SEEK3850
      B=0.E0                                                            SEEK3860
      K=0                                                               SEEK3870
      DO 850 I=1,NDF                                                    SEEK3880
```

```
      DIAG(I)=DIAG(I)+C1*DELTAG(I)**2+C2*DELTAG(I)*GRADI(I)
      DEFOR(I)=0.E0
      DO 830 J=I,NDF
      K=K+1
  830 DEFOR(I)=DEFOR(I)+CURV(K)*DELTAG(J)
      IF (I.EQ.1) GO TO 850
      J1=I-1
      DO 840 J=1,J1
      L=NDF*(J-1)+I-J*(J-1)/2
  840 DEFOR(I)=DEFOR(I)+CURV(L)*DELTAG(J)
  850 B=B-DEFOR(I)*DELTAG(I)
      IF (IPRINT.GT.1) WRITE (NPRT,240) (DIAG(I),I=1,NDF)
      K=0
      DO 860 I=1,NDF
      DO 860 J=I,NDF
      K=K+1
  860 CURV(K)=CURV(K)+DIRECT(I)*DIRECT(J)/AVAL+DEFOR(I)*DEFOR(J)/B
C
C     CHECK THAT DIAGONAL ELEMENTS ARE POSITIVE
C
      DO 910 I=1,NDF
      K=NDF*(I-1)+I-I*(I-1)/2
      IF (CURV(K).GT.0.E0) GO TO 910
      WRITE (NPRT,870) I
  870 FORMAT (51H  * NEGATIVE DIAGONAL IN CURVATURE-INVERSE MATRIX,,I4,
     143H ROW ZEROED, DIAGONAL SET TO UNITY. (SEEK)*)
      CURV(K)=1.E0
      DIAG(I)=1.E0
      IF (NDF.EQ.1) GO TO 910
      IF (I.EQ.NDF) GO TO 890
```

```
      J1=I+1                                                            SEEK4190
      DO 880  J=J1,NDF                                                  SEEK4200
      K=K+1                                                             SEEK4210
880   CURV(K)=0.E0                                                      SEEK4220
890   IF (I.EQ.1) GO TO 910                                             SEEK4230
      J1=I-1                                                            SEEK4240
      DO 900  J=1,J1                                                    SEEK4250
      L=NDF*(J-1)+I-J*(J-1)/2                                           SEEK4260
900   CURV(L)=0.E0                                                      SEEK4270
910   CONTINUE                                                          SEEK4280
C                                                                       SEEK4290
C     GO BACK FOR ANOTHER ITERATION                                     SEEK4300
C                                                                       SEEK4310
      GO TO 250                                                         SEEK4320
C                                                                       SEEK4330
C     RETURN TO CALLING PROGRAM                                         SEEK4340
C                                                                       SEEK4350
920   IF (IPRINT.GT.0) GO TO 940                                        SEEK4360
      WRITE (NPRT,110) ENGY,(DEFORM(I),I=1,NDF)                         SEEK4370
      WRITE (NPRT,610) KOUNT,NCOUNT                                     SEEK4380
      WRITE(NPRT,930) KLIN                                              SEEK4390
930   FORMAT (23H  LINEAR MINIMIZATIONS=,I10)                           SEEK4400
      WRITE (NPRT,690) EMAX                                             SEEK4410
940   RETURN                                                            SEEK4420
C                                                                       SEEK4430
C ***** GLOSSARY FOR SEEK *****                                         SEEK4440
C                                                                       SEEK4450
C AVAL   = ESTIMATE OF SIGN OF CURVATURE; IF POSITIVE, PARABOLA OPENS   SEEK4460
C          UPWARD, IF NEGATIVE PARABOLA OPENS DOWNWARD.                 SEEK4470
C CURV   = INVERSE MATRIX OF SECOND ORDER PARTIAL DERIVATIVES           SEEK4480
```

```
C          (CURVATURES) OF THE FUNCTION.                                   SEEK4490
C DEFOR  = NEW ESTIMATE OF DEFORMATION VECTOR.                              SEEK4500
C DEFORM = DEFORMATION VECTOR.                                              SEEK4510
C DELTAG = VECTOR OF CHANGES IN THE GRADIENT.                               SEEK4520
C DIAG   = SECOND DERIVATIVES OF THE FUNCTION ALONG COORDINATE              SEEK4530
C          DIRECTIONS (MAIN DIAGONAL OF CURVATURE MATRIX).                  SEEK4540
C DIRECT = DIRECTION VECTOR.                                                SEEK4550
C DISACC = ACCURACY REQUIRED OF THE DISPLACEMENT COMPONENTS.                SEEK4560
C DSTEP  = DIFFERENCE OF TWO SUCCESSIVE STEPSIZES.                          SEEK4570
C DX     = DIRECTIONAL DERIVATIVE OF THE FUNCTION ALONG THE LINE;           SEEK4580
C          SCALAR PRODUCT OF THE GRADIENT VECTOR AND THE DIR. VECTOR.       SEEK4590
C EMAX   = MAXIMUM CHANGE IN VALUE OF ALL DISPLACEMENTS.                    SEEK4600
C ENGG   = ENERGY GRADIENT FOR ONE DEGREE-OF-FREEDOM.                       SEEK4610
C ENGY   = VALUE OF THE POTENTIAL FUNCTION.                                 SEEK4620
C EPVAL  = ACCURACY MEASURE OF EVAL BASED ON DISACC.                        SEEK4630
C EQVAL  = ACCURACY MEASURE OF EVAL BASED ON MAGNITUDE OF DEFORM.           SEEK4640
C ETAM   = MEASURE OF ENERGY IMBALANCE.                                     SEEK4650
C ETEST  = MEASURE OF MAXIMUM STEPSIZE.                                     SEEK4660
C FUNACC = ESTIMATE OF RELATIVE ERROR IN FUNCTION EVALUATIONS.              SEEK4670
C FUNMIN = A LOWER BOUND ON THE FUNCTION.                                   SEEK4680
C FVAL   = FUNCTION VALUE.                                                  SEEK4690
C GRAD   = CURRENT GRADIENT VECTOR.                                         SEEK4700
C GRADI  = PREVIOUS GRADIENT VECTOR.                                        SEEK4710
C GRADM  = SQUARE OF THE MAGNITUDE OF THE GRADIENT VECTOR.                  SEEK4720
C IDENT  = FLAG TO IDENTIFY WHETHER OR NOT STEPSIZE SHOULD BE UPDATED.      SEEK4730
C IE     = SUBSCRIPT IDENTIFYING THE VALUE OF EVAL CLOSEST TO THE MIN.      SEEK4740
C IINITD = INDEX FOR INITIAL GUESS OF DISPLACEMENTS.                        SEEK4750
C ILOOP  = NUMBER OF TIMES A ZERO INITIAL GRADIENT IS FOUND.                SEEK4760
C INIT   = INDEX IDENTIFYING LINEAR SEARCH STEP.                            SEEK4770
C KKK    = COUNT OF NUMBER OF STEPS TAKEN ALONG LINE.                       SEEK4780
```

```
C KLIN   = NUMBER OF MINIMIZATIONS.                                    SEEK4790
C KCUNT  = NUMBER OF CALLS TO POTENTIAL FUNCTION ROUTINE.              SEEK4800
C MINIM  = SUBSCRIPT IDENTIFYING MINIMUM VALUE OF FVAL.                SEEK4810
C NCOUNT = NUMBER OF CALLS TO CHANGE IN POTENTIAL ROUTINE.             SEEK4820
C NLIN   = MAXIMUM NUMBER OF LINEAR MINIMIZATIONS TO BE PERFORMED.     SEEK4830
C SSIZE  = INITIAL STEPSIZE FOR APPROXIMATING DERIVATIVES.             SEEK4840
C STEP   = STEP SIZE IN DIRECTION OF LINEAR MINIMIZATION.              SEEK4850
C STEP2  = ESTIMATE OF SECOND STEPSIZE IN DIR. OF LINEAR MINIMIZATION. SEEK4860
C STEP3  = THIRD ESTIMATE OF STEPSIZE TO LOCATE MINIMUM.               SEEK4870
C STPSIZ = STEPSIZE USED TO APPROXIMATE DERIVATIVES BY DIFFERENCES.    SEEK4880
C                                                                      SEEK4890
C                                                                      SEEK4900
      END                                                              SEEK4910
```

```
CSTEN    0  10
      SUBROUTINE STEN(M,UR,UD,IFLAG)
C
C THIS SUBROUTINE CALCULATES THE CONTRIBUTION OF THE STEEL
C REINFORCEMENT TO THE TOTAL MEMBER ENERGY.
C
      COMMON DATA(10000),KDATA(500)
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9),
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20),
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,
     4    NTIMES,NVEL,IINITD
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45),
     5    YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,
     1    LTABI,NMAX,NMAXI
      COMMON/STRNBK/SRP(4),SRQ(4),UX,UY,UZ,XLEN,AREA,ZZI,IMAT
      INTEGER HEAD,DHEAD
C
```

```
      DIMENSION STRAIN(5),URI(5),UDI(5),GAUSS(3)
C
C INITIALIZE
      NG = NGRP(M)
      ISTAT=MSTAT(M)
      GAUSS(1)=5.E0/9.E0
      GAUSS(2)=8.E0/9.E0
      GAUSS(3)=GAUSS(1)
C RETRIEVE BEGINNING (ZERO) INDEXES FOR STRAIN AND STRESS HISTORIES.
      KRAIN=KDATA(LPI+M)-1
      KRESS=KDATA(LPSI+M)-1
C
C BEGIN DO LOOP OVER GROUP NUMBER.
C
      DO 70 I=1,NG
      MATL=MBAR(I,M)
      AG=AGRP(I,M)
      YLOC=YBAR(I,M)
      KS=KRESS+24*(I-1)
      KR=KRAIN+3*(I-1)
      CON = AG*XLEN/2.E0
C
C DETERMINE ENERGY DENSITY AT INTEGRATION SECTIONS.
      DO 60 J=1,3
      XLOC=XPI(J,M)
C RETRIEVE STRAIN AND STRESS HISTORY.
      IF(ISTAT.EQ.3) GO TO 18
      DO 16 L=1,9
16    S(L)=0.E0
      GO TO 25
```

```
18    LS=KS+8*(J-1)                                            STEN 590
      S(1)=DATA(LS+1)                                          STEN 600
      IF(S(1).EQ.1.E0) GO TO 30                                STEN 610
      DO 20 L=2,8                                              STEN 620
20    S(L)=DATA(LS+L)                                          STEN 630
      S(9)=DATA(KR+J)                                          STEN 640
C FIND STRAIN AT INTEGRATION SECTION.                           STEN 650
25    CALL STRN(M,XLOC,YLOC,STRAIN(J))                         STEN 660
C OBTAIN STEEL ENERGY DENSITY.                                  STEN 670
      CALL REIN(MATL,STRAIN(J),URI(J),UDI(J))                  STEN 680
      GO TO 40                                                 STEN 690
C IF FIBER HAS FAILED, SET ENERGY DENSITY TO ZERO.              STEN 700
30    URI(J)=0.E0                                              STEN 710
      UDI(J)=0.E0                                              STEN 720
      GO TO 65                                                 STEN 730
C UPDATE STRAIN AND STRESS HISTORY IF IFLAG=3.                  STEN 740
40    IF(IFLAG.NE.3.OR.ISTAT.NE.3) GO TO 65                    STEN 750
      DO 50 L=1,8                                              STEN 760
50    DATA(LS+L)=S(L)                                          STEN 770
      DATA(KR+J)=S(9)                                          STEN 780
C     ACCUMULATE ENERGY FOR THIS GROUP.                         STEN 790
65    UR = UR + CON*GAUSS(J)*URI(J)                            STEN 800
      UD = UD + CON*GAUSS(J)*UDI(J)                            STEN 810
60    CONTINUE                                                 STEN 820
70    CONTINUE                                                 STEN 830
      IF (IFLAG.NE.3.OR.ISTAT.NE.3) GO TO 110                  STEN 840
       KR=KRAIN+3*NG                                           STEN 850
       KS=KRESS+24*NG                                          STEN 860
       XLOC=0.E0                                               STEN 870
      DO 100 I=1,2                                             STEN 880
```

```
      IF(I.EQ.2) XLOC=XLEN
       J=I
       DO 90 K=1,NG
       CALL STRN(M,XLOC,YBAR(K,M),TRAIN)
      IF(ABS(TRAIN).LT.TINY) TRAIN=TINY
      S(9)=DATA(KR+J)
      LS=KS+8*(J-1)
      DO 75 L=1,8
75    S(L)=DATA(LS+L)
       CALL REIN(MATL,TRAIN,URE,UDE)
       DATA(KR+J)=S(9)
       DO 80 L=1,8
80     DATA(LS+L)=S(L)
       J=J+2
90     CONTINUE
100    CONTINUE
110    RETURN
C
C ***** GLOSSARY FOR STEN *****
C
C     AG = AREA OF LONGITUDINAL STEEL GROUP.
C    CON = TEMPORARY CONSTANT.
C      I = LONGITUDINAL STEEL GROUP NUMBER.
C      J = INTEGRATION SECTION NUMBER.
C     KR = STRAIN INDEX FOR EACH STEEL GROUP.
C  KRAIN = ZERO INDEX FOR STRAIN HISTORY STORAGE.
C   KRES = ZERO INDEX FOR STRESS HISTORY STORAGE.
C     KS = STRESS INDEX FOR EACH STEEL GROUP.
C     LS = STRESS INDEX FOR EACH INTEGRATION SECTION.
C      M = MEMBER NUMBER.
```

```
C  MATL = MATERIAL NUMBER OF LONGITUDINAL STEEL GROUP.                  STEN1190
C    NG = NUMBER OF LONGITUDINAL STEEL GROUPS IN MEMBER.                STEN1200
C    UD = DISSIPATIVE ENERGY DENSITY FOR MEMBER.                        STEN1210
C    UR = RECOVERABLE ENERGY DENSITY FOR MEMBER.                        STEN1220
C  XLEN = LENGTH OF MEMBER.                                             STEN1230
C  YLOC = DISTANCE FROM REFERENCE AXIS TO STEEL GROUP.                  STEN1240
C                                                                       STEN1250
      END                                                               STEN1260
```

```
CSTOR    0  10
      SUBROUTINE STOR (M)
C
C     THIS SUBROUTINE ESTABLISHES STORAGE SPACE FOR STRESS-STRAIN
C     HISTORY OF AN INELASTIC MEMBER.
C
      COMMON DATA(10000),KDATA(500)
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,
     4    NTIMES,NVEL,IINITD
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,
     1    LTABI,NMAX,NMAXI
C     COUNT NO. OF ELEMENTS WITH MSTAT = 3.
      IYLD = IYLD + 1
C     CALCULATE STRAIN AND STRESS HISTORY STORAGE REQUIREMENTS.
      L=LMAX+1
      LSTN=5*NGRP(M)+58
      IF (MTYPE(M).EQ.4) LSTN=43
      LSTS=8*LSTN
      LMAX=LMAX+LSTN+LSTS
C
C     COMPARE WITH AVAILABLE STORAGE.
      IF (LMAX.LE.NMAX) GO TO 20
      PRINT 10
10    FORMAT (//1H ,62H*** PROBLEM STORAGE REQUIREMENTS EXCEED DATA ARRA
```

```
     1Y (STOR). ***)                                                    STOR 290
      LERR=1                                                            STOR 300
      GO TO 30                                                          STOR 310
C                                                                       STOR 320
C     STORE ADDRESSES OF STRAIN AND STRESS HISTORY IN KDATA.            STOR 330
20    K=LPI+M                                                           STOR 340
      KDATA(K)=L                                                        STOR 350
      KK=LPSI+M                                                         STOR 360
      KDATA(KK)=L+LSTN                                                  STOR 370
C                                                                       STOR 380
C     INITIALIZE DATA VECTOR FOR MEMBER.                                STOR 390
      DO 40 K=L,LMAX                                                    STOR 400
   40 DATA(K)=0.E0                                                      STOR 410
C                                                                       STOR 420
C     RETURN TO CALLING PROGRAM.                                        STOR 430
C                                                                       STOR 440
30    RETURN                                                            STOR 450
C                                                                       STOR 460
C ***** GLOSSARY FOR STOR *****                                         STOR 470
C                                                                       STOR 480
C  LMAX = CURRENT LENGTH OF DATA VECTOR.                                STOR 490
C   LPI = STARTING INDEX IN KDATA OF STRAIN DATA ADDRESSES.             STOR 500
C  LPSI = STARTING INDEX IN KDATA OF STRESS DATA ADDRESSES.             STOR 510
C  LSTN = LENGTH OF STRAIN VECTOR IN DATA ARRAY FOR MEMBER.             STOR 520
C  LSTS = LENGTH OF STRESS VECTOR IN DATA ARRAY FOR MEMBER.             STOR 530
C  NMAX = DIMENSIONED LENGTH OF DATA VECTOR.                            STOR 540
C                                                                       STOR 550
      END                                                               STOR 560
```

```
CSTRN    0  10
      SUBROUTINE STRN (M,X,Y,STRAIN)
C
C SUBROUTINE TO CALCULATE LONGITUDINAL STRAIN AT ANY POINT
C WITHIN THE MACRO ELEMENT BASED ON LOCAL DISPLACEMENTS.
C
      COMMON /MEMBER/ DUMMY1(5365),XDM(45),DUMMY2(360)
      COMMON/STRNBK/SRP(4),SRQ(4),UX,UY,UZ,XLEN,AREA,ZZI,IMAT
C
      XI = X/XLEN
      ETA = Y/XLEN
C
C     DERIVATIVES OF DISPLACEMENT FUNCTIONS.
      P1P = 4.E0*XI - 1.E0
      P4P = 4.E0 - 8.E0*XI
      P2P = 6.E0*(XI-XI**2)
      P3P = 3.E0*XI**2 - 2.E0*XI
      P2PP = 6.E0 - 12.E0*XI
      P3PP = 6.E0*XI - 2.E0
C
C     NORMAL STRAIN AT ANY POINT X,Y (XI,ETA).
      STRAIN = P1P*UX + P4P*XDM(M) + .5E0*(P2P*UY+P3P*UZ)**2
     1         - ETA*(P2PP*UY+P3PP*UZ)
      RETURN
C
C     ***** GLOSSARY FOR STRN *****
C
C    ETA = NONDIMENSIONAL Y LOCATION.
C      M = MEMBER OR ELEMENT NUMBER.
C    P1P = PHI-ONE-PRIME, DERIVATIVE OF PHI1 W.R.T. X - THE 1
```

```
C           CORRESPONDS TO THE X-DIRECTION, 2 TO Y, 3 TO Z OR ROTATION,  STRN 290
C           AND 4 CORRESPONDS TO THE INTERNAL NODE.                      STRN 300
C  STRAIN = NORMAL STRAIN AT X,Y WITHIN THE ELEMENT.                     STRN 310
C      UX = LOCAL DISPLACEMENT IN THE X OR 1 DIRECTION.                  STRN 320
C       X = X COORDINATE WITHIN THE ELEMENT.                             STRN 330
C     XDM = LONGITUDINAL DISPLACEMENT OF THE INTERNAL NODE.              STRN 340
C      XI = NONDIMENSIONAL X LOCATION.                                   STRN 350
C    XLEN = LENGTH OF THE ELEMENT.                                       STRN 360
C       Y = Y COORDINATE WITHIN THE ELEMENT.                             STRN 370
C                                                                        STRN 380
      END                                                                STRN 390
```

```
CSUMY    0  10
      SUBROUTINE SUMY                                                   SUMY   0
C     THIS SUBROUTINE PRINTS A SUMMARY OF PROBLEM CHARACTERISTICS, AS INSUMY  10
C     BY SINGER                                                         SUMY  20
C                                                                       SUMY  30
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),       SUMY  40
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),SUMY  50
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)                      SUMY  60
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9), SUMY  70
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)SUMY  80
      COMMON/JOINTS/ACC(3,50),BET(3,50),DAS(3,50),DIS(3,50),ERJF(3,50), SUMY  90
     1    ERJH(3,50),ERJZ(3,50),F(3,50),FOR(3,50),VEL(3,50),X(50),      SUMY 100
     1    XDJ(3,50),Y(50),DER(3,50),RESENG(3,50),IDFI(90),IDFII(90)     SUMY 110
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20), SUMY 120
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB      SUMY 130
      INTEGER HEAD,DHEAD                                                SUMY 140
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,SUMY 150
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,SUMY 160
     2    NCRO,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, SUMY 170
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  SUMY 180
     4    NTIMES,NVEL,IINITD                                            SUMY 190
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),SUMY 200
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),    SUMY 210
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),SUMY 220
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                SUMY 230
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45), SUMY 240
     5    YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)              SUMY 250
C     PRINT THE PAGE NUMBER                                             SUMY 260
C                                                                       SUMY 270
      CALL PAGE                                                         SUMY 280
```

```
10    FORMAT (1H ,30X,34H CHARACTERISTICS OF THE SIMULATION,//)           SUMY 290
      WRITE (NPRT,10)                                                     SUMY 300
C     PRINT THE TABLE HEADING                                             SUMY 310
C                                                                         SUMY 320
20    FORMAT (1H0,10HITEM CLASS,8X,16HITEM DESCRIPTION)                   SUMY 330
      WRITE (NPRT,20)                                                     SUMY 340
30    FORMAT(1H ,11(1H-),7X,70(1H-)//)                                    SUMY 350
      WRITE (NPRT,30)                                                     SUMY 360
C                                                                         SUMY 370
C     PRINT THE PARAMETERS OF THE STRUCTURE.                              SUMY 380
C                                                                         SUMY 390
40    FORMAT (11H PARAMETERS,9X,I5,7H JOINTS,/,20X,I5,9H ELEMENTS,/,7H OSUMY 400
     1F THE,13X,I5,10H MATERIALS,/,20X,I5,13H POINT MASSES,/,10H STRUCTUSUMY 410
     2RE,///)                                                             SUMY 420
      LINE=LINE+9                                                         SUMY 430
      NEL=NM+NLS                                                          SUMY 440
      WRITE (NPRT,40) NJ,NEL,NMAT,NMAS                                    SUMY 450
C                                                                         SUMY 460
C     PRINT KINEMATIC CONDITIONS.                                         SUMY 470
C                                                                         SUMY 480
50    FORMAT (10H KINEMATIC,10X,I5,27H DEGREES OF FREEDOM (TOTAL)/20X,    SUMY 490
     1I5,27H ELEMENT DEGREES OF FREEDOM/20X,I5,33H LEAF SPRING RIGIDITY SUMY 500
     2CONSTRAINTS/11H CONDITICNS,9X,I5,34H DISPLACEMENTS PRESCRIBED AS ZSUMY 510
     3ERO ,///)                                                           SUMY 520
      LINE=LINE+8                                                         SUMY 530
      IF (LINE.GE.NL) CALL PAGE                                           SUMY 540
      NRC=NJ*3-NDFJ                                                       SUMY 550
      WRITE (NPRT,50) NDF,NM,NLSR,NRC                                     SUMY 560
C                                                                         SUMY 570
C     PRINT FORCE CONDITIONS.                                             SUMY 580
```

```
C
60    FORMAT (6H FORCE,14X,I5,25H JOINT LOADING CONDITIONS,/,20X,I5,21H
     1DISTRIBUTED LOADINGS,/,11H CONDITIONS,9X,I5,16H FUNCTION TABLES,//
     2/)
      LINE=LINE+6
      IF (LINE.GE.NL) CALL PAGE
      WRITE (NPRT,60) NFF,NDL,NTAB
C
C     PRINT INITIAL CONDITION DATA
C
70    FORMAT (8H INITIAL,12X,I5,32H JOINTS WITH DISPLACEMENTS GIVEN/,20X
     1,I5,30H JOINTS WITH VELOCITIES GIVEN ,/,11H CONDITION ,9X,I5,33H J
     2OINTS WITH ACCELERATIONS GIVEN ,/,20X,I5,24H JOINTS WITH JERKS GIV
     3EN,/,6H DATA ,14X,I5,31H JOINTS WITH POINT FORCES GIVEN///)
      WRITE (NPRT,70) NDIS,NVEL,NACC,NJER,NJOR
      LINE=LINE+8
      IF (LINE.GE.NL) CALL PAGE
C
C     PRINT SIMULATION ASSUMPTIONS
C
80    FORMAT (12H ASSUMPTIONS,8X,23H TIME-HISTORY STARTS AT,0PE12.4,9H S
     1ECONDS.,/,1H ,20X,12HIT STOPS AT ,0PE12.4,9H SECONDS.,/,1H ,19X,44
     2H MAXIMUM TOLERABLE RELATIVE ENERGY ERROR IS ,0PE12.4,1H.,/,7H OF 
     3THE,13X,48H THE MAXIMUM COMPUTER RUN TIME FOR THIS CASE IS ,0PE12.
     44,9H MINUTES.)
      LINE=LINE+12
      IF (LINE.GE.NL) CALL PAGE
      WRITE (NPRT,80) TBEGIN,THALT,SERR,TPROB
      IF (IUNITS-1) 100,120,130
90    FORMAT (1H ,19X,53H INPUT IS IN ENGLISH UNITS, OUTPUT IS IN ENGLIS
```

```
      1H TOO.)                                                          SUMY 890
100   WRITE (NPRT,90)                                                   SUMY 900
      GO TO 180                                                         SUMY 910
110   FORMAT (1H ,19X,70H INPUT IS IN ENGLISH UNITS, OUTPUT IS IN STANDASUMY 920
     1RD INTERNATIONAL UNITS.)                                          SUMY 930
120   WRITE (NPRT,110)                                                  SUMY 940
      GO TO 180                                                         SUMY 950
130   IF (IUNITS-2) 120,150,170                                         SUMY 960
140   FORMAT (1H ,19X,70H INPUT IS IN STANDARD INTERNATIONAL UNITS, OUTPSUMY 970
     1UT IS IN STANDARD TOO. )                                          SUMY 980
150   WRITE (NPRT,140)                                                  SUMY 990
      GO TO 180                                                         SUMY1000
160   FORMAT (1H ,19X,70H INPUT IS IN STANDARD INTERNATIONAL UNITS, OUTPSUMY1010
     1UT IS IN ENGLISH UNITS.)                                          SUMY1020
170   WRITE (NPRT,160)                                                  SUMY1030
180   IF(ILIN.EQ.1) GO TO 220                                           SUMY1040
190   FORMAT (1H ,20X,73HJOINT ROTATIONS AND ELEMENT DISTORTIONS ARE ASSSUMY1050
     1UMED TO BE INFINITESIMALS.)                                       SUMY1060
      WRITE (NPRT,190)                                                  SUMY1070
      GO TO 280                                                         SUMY1080
210   FORMAT (1H ,20X,70HJOINT ROTATIONS ARE ASSUMED FINITE, ELEMENT DISSUMY1090
     1TORTIONS INFINITESIMAL.)                                          SUMY1100
220   WRITE (NPRT,210)                                                  SUMY1110
      GO TO 280                                                         SUMY1120
280   IF (ISTOP) 320,320,300                                            SUMY1130
290   FORMAT (11H SIMULATION,10X,63HCALCULATIONS ARE STOPPED ON POTENTIASUMY1140
     1LLY FATAL AND FATAL ERRORS.)                                      SUMY1150
300   WRITE (NPRT,290)                                                  SUMY1160
      GO TO 340                                                         SUMY1170
310   FORMAT (11H SIMULATION,9X,46HCALCULATIONS ARE STOPPED ONLY ON FATASUMY1180
```

```
    1L ERRORS.)                                                          SUMY1190
320  WRITE (NPRT,310)                                                     SUMY1200
330  FORMAT (1H ,20X,37HSHEAR MACRO FAILURE IS DEFINED BY CA=,0PE12.4,5SUMY1210
    1H, CB=,0PE12.4,1H,/,1H ,22X,3HCC=,0PE12.4,5H CD= ,0PE12.4,        SUMY1220
    29H, AND CE=,0PE12.4,1H.///)                                         SUMY1230
340  WRITE (NPRT,330) CA,CB,CC,CD,CE                                      SUMY1240
C                                                                         SUMY1250
C     PRINT SIMULATION OPTIONS.                                           SUMY1260
C                                                                         SUMY1270
      IF (ISTART.GT.0) GO TO 370                                          SUMY1280
350   FORMAT (8H OPTIONS,13X,31HINPUT DATA IS IN PUNCHED CARDS.)          SUMY1290
      LINE=LINE+9                                                         SUMY1300
      IF (LINE.GE.NL) CALL PAGE                                           SUMY1310
      WRITE (NPRT,350)                                                    SUMY1320
      GO TO 380                                                           SUMY1330
360   FORMAT (8H OPTIONS,13X,21HINPUT DATA IS ON UNIT,I5)                 SUMY1340
370   IST=11                                                              SUMY1350
      WRITE (NPRT,360) IST                                                SUMY1360
380   IF (IPRINT) 400,400,430                                             SUMY1370
390   FORMAT (1H ,20X,48HTHE +MINIMUM+ OF SIMULATION RESULTS IS PRINTED.SUMY1380
     1 )                                                                  SUMY1390
400   WRITE (NPRT,390)                                                    SUMY1400
      GO TO 480                                                           SUMY1410
410   FORMAT (1H ,20X,45HTHE +STANDARD+ AMOUNT OF RESULTS IS PRINTED. ) SUMY1420
420   WRITE (NPRT,410)                                                    SUMY1430
      GO TO 480                                                           SUMY1440
430   IF (IPRINT-2) 420,450,470                                           SUMY1450
440   FORMAT (1H ,20X,44H+DETAILS+ OF SIMULATION RESULTS ARE PRINTED.)  SUMY1460
450   WRITE (NPRT,440)                                                    SUMY1470
      GO TO 480                                                           SUMY1480
```

```
460   FORMAT (1H ,20X,50HALL PRINT STATEMENTS CITING RESULTS ARE EXECUTESUMY1490
     1D. )                                                               SUMY1500
470   WRITE (NPRT,460)                                                   SUMY1510
480   IF (ISTRES) 500,520,530                                            SUMY1520
490   FORMAT (8H FOR THE,13X,39HPRINTOUTS EXCLUDE STRESSES AND STRAINS.)SUMY1530
500   WRITE (NPRT,490)                                                   SUMY1540
      GO TO 590                                                          SUMY1550
510   FORMAT (8H FOR THE,13X,39HPRINTOUTS INCLUDE STRESSES AND STRAINS.)SUMY1560
520   WRITE (NPRT,510)                                                   SUMY1570
      GO TO 590                                                          SUMY1580
530   IF (ISTRES-2) 520,550,570                                          SUMY1590
540   FORMAT (8H FOR THE,13X,36HPRINTOUTS INCLUDE STRESS RESULTANTS.)    SUMY1600
550   WRITE (NPRT,540)                                                   SUMY1610
      GO TO 590                                                          SUMY1620
560   FORMAT (8H FOR THE,13X,59HPRINTOUTS INCLUDE STRESSES, STRAINS, ANDSUMY1630
     1 STRESS RESULTANTS.)                                               SUMY1640
570   WRITE (NPRT,560)                                                   SUMY1650
580   FORMAT (1H ,20X,21HPRINTOUTS OCCUR EVERY,0PE12.4,35H SECONDS, OR LSUMY1660
     1ESS, OF THE HISTORY. )                                             SUMY1670
590   WRITE (NPRT,580) TINK                                              SUMY1680
      IF (ITAPE.EQ.0) GO TO 620                                          SUMY1690
600   FORMAT (15H INPUT ^ OUTPUT,6X,37HCONTINUATION DATA IS WRITTEN ON USUMY1700
     1NIT ,I2)                                                           SUMY1710
      IST=11                                                             SUMY1720
      WRITE (NPRT,600) IST                                               SUMY1730
      GO TO 630                                                          SUMY1740
610   FORMAT (15H INPUT ^ OUTPUT,6X,38HNO CONTINUATION DATA FILE IS WRITSUMY1750
     1TEN. )                                                             SUMY1760
620   WRITE (NPRT,610)                                                   SUMY1770
630   IF (IPLOT.EQ.0) GO TO 650                                          SUMY1780
```

```
640   FORMAT (1H ,20X,44HA DATA RETRIEVAL FILE IS WRITTEN ON FILE 10.)  SUMY1790
      WRITE (NPRT,640)                                                  SUMY1800
650   CONTINUE                                                          SUMY1810
C                                                                       SUMY1820
C     NOTE AVAILABLE STORAGE                                            SUMY1830
C                                                                       SUMY1840
      LINE=LINE+9                                                       SUMY1850
      IF (LINE.GE.NL) CALL PAGE                                         SUMY1860
      NA=NJD-NJ                                                         SUMY1870
      NB=NMD-NM                                                         SUMY1880
      NC=NMATD-NMAT                                                     SUMY1890
      ND=NDFD-NDF                                                       SUMY1900
660   FORMAT (1H0,8X,40H STORAGE STILL AVAILABLE FOR MODELING   /11X,I5,SUMY1910
     17H JOINTS /11X,I5,9H ELEMENTS /11X,I5,10H MATERIALS /11X,I5,9H FRESUMY1920
     2EDOMS ////)                                                       SUMY1930
      WRITE (NPRT,660) NA,NB,NC,ND                                      SUMY1940
      RETURN                                                            SUMY1950
      END                                                               SUMY1960
```

```
CTABL    0  10
      SUBROUTINE TABL (N,TX,J,VALUE)
C
C     THIS SUBROUTINE EVALUATES A FORCING FUNCTION FROM THE TABLE OF FOR
C     FUNCTIONS
C
C          N      NUMBER OF FORCING FUNCTION
C          TX     TIME FOR WHICH FUNCTION WILL BE EVALUATED
C          J      NUMBER OF TIME POINT USED FOR LAST EVALUATION
C          VALUE  VALUE OF FORCING FUNCTION
C
      COMMON DATA(10000),KDATA(500)
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,
     1    LTABI,NMAX,NMAXI
C
C     DETERMINE LOCATION OF FIRST TIME POINT IN FORCING FUNCTION
C
      LFTP=KDATA(LTABI+2*N-1)
C
C     DETERMINE NUMBER OF TIME POINTS IN FORCING FUNCTION
C
      NTP=KDATA(LTABI+2*N)
C
C     DETERMINE LOCATION OF LAST TIME POINT IN FORCING FUNCTION
C
      LLTP=LFTP+NTP*2-2
C
C     IF TIME FOR EVALUATION IS LESS THAN FIRST TIME POINT, FUNCTION EQU
C     AT FIRST TIME POINT
C
```

```
      IF (TX.GT.DATA(LFTP)) GO TO 10                                    TABL 290
      VALUE=DATA(LFTP+1)                                                TABL 300
      GO TO 70                                                          TABL 310
C                                                                       TABL 320
C     IF TIME FOR EVALUATION IS GREATER THAN LAST TIME POINT, FUNCTION ETABL 330
C     FORCE AT LAST TIME POINT                                          TABL 340
C                                                                       TABL 350
10    IF (TX.LT.DATA(LLTP)) GO TO 20                                    TABL 360
      VALUE=DATA(LLTP+1)                                                TABL 370
      GO TO 70                                                          TABL 380
C                                                                       TABL 390
C     EVALUATE FUNCTION FOR EVALUATION TIME BETWEEN FIRST AND LAST TIME TABL 400
C                                                                       TABL 410
20    LJTP = LFTP + 2*(J-1)                                             TABL 420
      IF (TX.LE.DATA(LJTP)) GO TO 40                                    TABL 430
30    J=J+1                                                             TABL 440
      LJTP=LJTP+2                                                       TABL 450
      IF (TX.LE.DATA(LJTP)) GO TO 50                                    TABL 460
      GO TO 30                                                          TABL 470
40    IF (TX.GE.DATA(LJTP-2)) GO TO 50                                  TABL 480
      J=J-1                                                             TABL 490
      LJTP=LJTP-2                                                       TABL 500
      GO TO 40                                                          TABL 510
50    TJ=DATA(LJTP)                                                     TABL 520
      FJ=DATA(LJTP+1)                                                   TABL 530
      TI=DATA(LJTP-2)                                                   TABL 540
      FI=DATA(LJTP-1)                                                   TABL 550
      DELT=TJ-TI                                                        TABL 560
      IF (DELT.EQ.0.E0) GO TO 60                                        TABL 570
      VALUE=FI+(FJ-FI)*(TX-TI)/DELT                                     TABL 580
```

```
      GO TO 70                                                          TABL 590
60    VALUE=FI                                                          TABL 600
70    RETURN                                                            TABL 610
      END                                                               TABL 620
```

```
CTEST    0  10
      SUBROUTINE TEST (IYFLAG)                                          TEST   0
C     THIS SUBROUTINE TESTS FOR NEWLY YIELDED MEMBERS OF MSTAT(M)=2     TEST  10
C     STATUS.  IF SUCH MEMBERS ARE FOUND, THEIR STATUS IS CHANGED FROM  TEST  20
C     ELASTIC TO PLASTIC FOR ALL FURTHER CALCULATIONS AND, UPON         TEST  30
C     RETURN TO THE CALLING PROGRAM, THE CURRENT SOLUTION IS REPEATED   TEST  40
C     WITH THESE MEMBERS PERMITTED TO YIELD.                            TEST  50
C                                                                       TEST  60
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),       TEST  70
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45).MSTAT(45),MTIES(45),TEST  80
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)                      TEST  90
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9), TEST 100
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)TEST 110
      COMMON/MAINBK/IANAL,ICURV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,TEST 120
     1    IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,TEST 130
     2    NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD, TEST 140
     3    NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,  TEST 150
     4    NTIMES,NVEL,IINITO                                            TEST 160
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),TEST 170
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),    TEST 180
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),TEST 190
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),                TEST 200
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45), TEST 210
     5    YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)              TEST 220
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,          TEST 230
     1    LTABI,NMAX,NMAXI                                              TEST 240
      COMMON/STRNBK/SRP(4),SRQ(4),UX,UY,UZ,XLEN,AREA,ZZI,IMAT           TEST 250
C                                                                       TEST 260
C     BEGIN DO LOOP OVER MEMBER NUMBER.                                 TEST 270
      DO 70 M=1,NM                                                      TEST 280
```

```
      IF (MSTAT(M).NE.2) GO TO 70                                       TEST 290
      L=MTYPE(M)                                                        TEST 300
      XLEN=XL(M)                                                        TEST 310
C                                                                       TEST 320
C     DETERMINE LOCAL MEMBER DEFORMATIONS.                              TEST 330
      CALL DEFO (M)                                                     TEST 340
C                                                                       TEST 350
C     CHECK EXTREME FIBER STRAINS AT ENDS OF MEMBER.                    TEST 360
C                                                                       TEST 370
      BEGM=XBEGM(M)                                                     TEST 380
      ENDM=EFLM(M)+BEGM                                                 TEST 390
C                                                                       TEST 400
C     TEST FOR STEEL WIDE FLANGE.                                       TEST 410
      IF (L.EQ.4) GO TO 20                                              TEST 420
      HT=HTOP(M)                                                        TEST 430
      HB=HT-HMEM(M)                                                     TEST 440
      MAT=MCODE(M)                                                      TEST 450
      GO TO 30                                                          TEST 460
   20 HT=HTWF(M)                                                        TEST 470
      HB=HT-DWF(M)                                                      TEST 480
      MAT=MATW(M)                                                       TEST 490
   30 EXSTN = ABS(STN(2,MAT))                                           TEST 500
      XDIST = BEGM                                                      TEST 510
      DO 35 I=1,2                                                       TEST 520
      IF(I.EQ.2) XDIST = ENDM                                           TEST 530
      HDIST = HT                                                        TEST 540
      DO 35 II=1,2                                                      TEST 550
      IF (II.EQ.2) HDIST = HB                                           TEST 560
      CALL STRN (M,XDIST,HDIST,EPST)                                    TEST 570
      IF(L.NE.4.AND.EPST.GE.0.E0) GO TO 35                              TEST 580
```

```
      IF(ABS(EPST).GT.EXSTN) GO TO 50                                   TEST 590
   35 CONTINUE                                                          TEST 600
C                                                                       TEST 610
C     CHECK FIBER STRAINS AT ENDS OF LONGITUDINAL STEEL GROUPS.         TEST 620
C                                                                       TEST 630
      IF (L.EQ.4) GO TO 70                                              TEST 640
      NG=NGRP(M)                                                        TEST 650
      DO 40 I=1,NG                                                      TEST 660
      MAT=MBAR(I,M)                                                     TEST 670
      STSTN=ABS(STN(2,MAT))                                             TEST 680
      BEG=XBEG(I,M)                                                     TEST 690
      END=BEG+EFFL(I,M)                                                 TEST 700
      YB=YBAR(I,M)                                                      TEST 710
      CALL STRN (M,BEG,YB,EPSB)                                         TEST 720
      IF (ABS(EPSB).GT.STSTN) GO TO 50                                  TEST 730
      CALL STRN (M,END,YB,EPSE)                                         TEST 740
      IF (ABS(EPSE).GT.STSTN) GO TO 50                                  TEST 750
   40 CONTINUE                                                          TEST 760
      GO TO 70                                                          TEST 770
C                                                                       TEST 780
C     PRINT NEWLY YIELDED MEMBER NUMBER, CHANGE MEMBER STATUS CODE,     TEST 790
C     AND ESTABLISH DATA STORAGE FOR STRAIN AND STRESS HISTORY.         TEST 800
C                                                                       TEST 810
   50 IF(IYFLAG.EQ.0) CALL PAGE                                         TEST 820
      IYFLAG = 1                                                        TEST 830
      MSTAT(M)=3                                                        TEST 840
      WRITE (NPRT,60) NAME(MAT),M                                       TEST 850
   60 FORMAT(10X,20HSTRAIN OF MATERIAL  ,A4,37H  EXCEEDS YIELD STRAIN, MTEST 860
     1EMBER NUMBER,I5,21H  HAS YIELDED (TEST).)                         TEST 870
      CALL STOR (M)                                                     TEST 880
```

```
C                                                                       TEST 890
C     END OF DO LOOP OVER MEMBER NUMBER.                                TEST 900
70    CONTINUE                                                          TEST 910
C                                                                       TEST 920
C     MESSAGE PRINTOUT AND RETURN TO CALLING PROGRAM.                   TEST 930
C                                                                       TEST 940
      IF (IYFLAG.NE.0) GO TO 90                                         TEST 950
      WRITE (NPRT,80)                                                   TEST 960
80    FORMAT (//1H ,42HTHERE ARE NO NEWLY YIELDED MEMBERS (TEST).)      TEST 970
      GO TO 170                                                         TEST 980
90    IF (LERR.EQ.0.AND.IERR.EQ.0) GO TO 110                            TEST 990
      PRINT 100                                                         TEST1000
100   FORMAT (1H ,80H*** THE ANALYSIS IS TERMINATED.  CURRENT RESULTS FOTEST1010
     1R THIS TIME STEP ARE PRINTED./5X,84HTHESE RESULTS ARE BASED ON LINTEST1020
     2EAR ELASTIC RESPONSE FOR THE ABOVE MEMBERS (TEST). ***)           TEST1030
      GO TO 170                                                         TEST1040
C                                                                       TEST1050
C     PRINT STORAGE LOCATION INDEXES.                                   TEST1060
110   WRITE (NPRT,120)                                                  TEST1070
120   FORMAT (//10X,46HSTORAGE LOCATION INDEXES IN DATA ARRAY (TEST)./) TEST1080
      WRITE (NPRT,130) LCURV,LTAB,LFF,LP,LMAX,NMAX                      TEST1090
130   FORMAT (14X,17HMINIMIZATION DATA,I19/14X,15HFUNCTION TABLES,I21/14TEST1100
     1X,17HFORCING FUNCTIONS,I19/14X,19HPLASTIC STRAIN DATA,I17/14X,12HETEST1110
     2ND OF ARRAY,I24/14X,16HSPACES ALLOCATED,I20//)                    TEST1120
      WRITE (NPRT,140)                                                  TEST1130
140   FORMAT (10X,47HSTORAGE LOCATION INDEXES IN KDATA ARRAY (TEST)./)  TEST1140
      WRITE (NPRT,150) LTABI,LFFI,LPI,LPSI,LMAXI,NMAXI                  TEST1150
150   FORMAT (14X,15HFUNCTION TABLES,I21/14X,17HFORCING FUNCTIONS,I19/14TEST1160
     1X,19HPLASTIC STRAIN DATA,I17/14X,19HSTRESS HISTORY DATA,I17/14X,12TEST1170
     2HEND OF ARRAY,I24/14X,16HSPACES ALLOCATED,I20//)                  TEST1180
```

```
      WRITE (NPRT,160)                                                  TEST1190
160   FORMAT (1H ,78HTHIS TIME STEP IS REPEATED WITH THE ABOVE MEMBERS TTEST1200
     1REATED AS INELASTIC (TEST).)                                      TEST1210
      IF (IYLD.LT.NM) GO TO 168                                         TEST1220
      WRITE (NPRT,164)                                                  TEST1230
164   FORMAT (1H ,38H***** ALL ELEMENTS HAVE YIELDED. *****)            TEST1240
168   CALL PAGE                                                         TEST1250
170   RETURN                                                            TEST1260
      END                                                               TEST1270
```

```
CTICS   0  10
      SUBROUTINE TICS(TIMN,IGA)
C
C     THIS IS A MACHINE DEPENDENT SUBROUTINE.
C
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20),
     1   PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB
      COMMON/MAINBK/IANAL,ICUFV,IERR,IFAIL,IFOR,ILIN,IPAGE,IPLOT,IPRINT,
     1   IREC,ISTART,ISTOP,ISTRES,ITAPE,IUNITS,IYLD,LERR,LINE,NACC,NCM,
     2   NCRD,NDF,NDFD,NDFJ,NDIS,NDL,NFF,NJOR,NINC,NJ,NJD,NJER,NL,NLD,
     3   NLS,NLSR,NM,NMAS,NMAT,NMATD,NMD,NPLOT,NPRT,NSAVE,NTAB,NTAPE,
     4   NTIMES,NVEL,IINITD
      COMMON /TIMEBK/ TCUM,INIT,KNT
      INTEGER HEAD,DHEAD
      IGA = 1
      IF(INIT.EQ.1) GO TO 20
C     INITIALIZE STARTING POINT
      CALL SECOND(T)
      INIT=1
      TT=0.E0
      GO TO 40
20    KNT = KNT + 1
      TIMN = (SECOND(TIMN) - T)/60.E0
      CALL SECOND (T)
      TCUM=TCUM+TIMN
      TPER=TCUM/KNT
      TEX=TCUM+TPER
      IF(TEX.GE.TPROB.OR.TIME.GT.THALT) IGA=2
      IF(TEX.GE.TPROB) PRINT 30
30    FORMAT(1H ,63H***ESTIMATED CALCULATION TIME EXCEEDS ALLOWED PROBLE
```

```
     1M (TICS)***)                                                      TICS 290
      TT=TT+TIMN                                                        TICS 300
      IF(TT.GE.15.0E0) GO TO 50                                         TICS 310
   40 RETURN                                                            TICS 320
   50 CALL REGO(-ISTART)                                                TICS 330
      TT=0.E0                                                           TICS 340
      GO TO 40                                                          TICS 350
      END                                                               TICS 360
```

```
CWIDE    0  10
      SUBROUTINE WIDE (M,UR,UD,IFLAG)
C     THIS SUBROUTINE GENERATES STRESS AND ENERGY CALCULATIONS FOR A
C     WIDE FLANGE STEEL MEMBER(M).  THE CALCULATIONS ARE CONTROLLED
C     BY (IFLAG),WHERE
C     IFLAG=1, INDICATES THAT THE RECOVERABLE STRAIN ENERGY(UR)
C              AND THE DISSIPATIVE STRAIN ENERGY (UD) ARE REQUIRED.
C     IFLAG=3, INDICATES THAT STRESS RESULTS ARE REQUIRED FOR PRINT-
C              OUT AND FOR UPDATING THE STRESS HISTORY DATA BANK.
C     THE ELASTIC-PLASTIC STATUS OF THE MEMBER IS INDICATED BY
C     ISTAT=MSTAT(M), WHERE
C     ISTAT=1, INDICATES A MEMBER THAT IS RESTRICTED TO LINEAR
C              ELASTIC RESPONSE.
C     ISTAT=2, INDICATES A MEMBER THAT IS CURRENTLY LINEAR ELASTIC
C              BUT MAY GO INELASTIC.
C     ISTAT=3, INDICATES A MEMBER THAT IS INELASTIC.
C
      COMMON DATA(10000),KDATA(500)
      COMMON/ELEMET/ICARD,IP(45),IPL(20),IQ(45),IQL(20),MATR(45),
     1    MATW(45),MBAR(10,45),MCODE(45),MSHEAR(45),MSTAT(45),MTIES(45),
     2    MTYPE(45),NGRP(45),NSPAC(6,45),NTIES(45)
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9),
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20),
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB
      INTEGER HEAD,DHEAD
      COMMON/MEMBER/AGRP(10,45),ATIES(6,45),BMEM(45),BPP(45),BDM(10,45),
     1    BWF(45),D(45),DP(45),DPP(45),DWF(45),EFFL(10,45),EFLM(45),
     2    HMEM(45),HTOP(45),HTWF(45),PDP(7,45),SPRING(5,20),STIES(7,45),
     3    TFWF(45),TWWF(45),UDM(45),URM(45),XBEG(10,45),
```

```
     4    XBEGM(45),XBEGS(6,45),XL(45),XPI(5,45),YBAR(10,45),YGP(7,45),
     5    YFIBR(11,45),YLDS(45),XDM(45),PDF(7,45),DISM(45)
      COMMON/STORE/LCURV,LFF,LFFI,LMAXI,LMAX,LP,LPI,LPSI,LTAB,
     1    LTABI,NMAX,NMAXI
      COMMON/STRNBK/SRP(4),SRQ(4),UX,UY,UZ,XLEN,AREA,ZZI,IMAT
C
      DIMENSION GPS(7,3),UDMP(7,3),URMP(7,3),GAUSS(3)
C
C INITIALIZE
C
      ISTAT = MSTAT(M)
      IMAT=MATW(M)
      XLEN=XL(M)
      GAUSS(1)=5.E0/9.E0
      GAUSS(2)=8.E0/9.E0
      GAUSS(3)=GAUSS(1)
      UR=0.E0
C DETERMINE LOCAL DEFORMATIONS
      CALL DEFO(M)
C CHECK FOR INELASTIC DEFORMATIONS.
      UD=0.E0
      IF(ISTAT.EQ.3) UD=UDM(M)
C OBTAIN ADDRESSES OF STRAIN AND STRESS HISTORY STORAGE.
      KRAIN=KDATA(LPI +M)-1
      KRESS=KDATA(LPSI+M)-1
C
C EVALUATE STRESSES AND ENERGY DENSITIES AT GAUSS POINTS.
      KR=KRAIN
      KS=KRESS
      DO 100 I=1,7
```

```
      DO 90 J=1,3
      CALL STRN(M,XPI(J,M),YGP(I,M),GPS(I,J))
      IF(ABS(GPS(I,J)).LT.TINY) GPS(I,J)=TINY
      IF(ISTAT.EQ.3) GO TO 50
      DO 40 L=1,9
40    S(L)=0.E0
      GO TO 70
50    S(9)=DATA(KR+J)
      LS=KS+8*(J-1)
      DO 60 L=1,8
60    S(L)=DATA(LS+L)
70    CALL REIN(IMAT,GPS(I,J),URMP(I,J),UDMP(I,J))
C UPDATE IF IFLAG=3.
      IF(IFLAG.NE.3.OR.ISTAT.NE.3) GO TO 90
      DATA(KR+J)=S(9)
      DO 80 L=1,8
80    DATA(LS+L)=S(L)
90    CONTINUE
      KR=KR+3
100   KS=KS+24
C
C OBTAIN ENERGY CONTRIBUTION OF FLANGES AND WEB.
C
       CONF=BWF(M)*TFWF(M)*XLEN/4.E0
       CONW=TWWF(M)*(DWF(M)-2.E0*TFWF(M))*XLEN/4.E0
       DO 110 I=1,2
       DO 110 J=1,3
       UR=UR+GAUSS(J)*URMP(I,J)*CONF
110    UD=UD+GAUSS(J)*UDMP(I,J)*CONF
       DO 120 I=1,3
```

```
      DO 120 J=1,3
      UR=UR+GAUSS(I)*GAUSS(J)*URMP(I+2,J)*CONW
120   UD=UD+GAUSS(I)*GAUSS(J)*UDMP(I+2,J)*CONW
      DO 130 I=6,7
      DO 130 J=1,3
      UR=UR+GAUSS(J)*URMP(I,J)*CONF
 130  UD=UD+GAUSS(J)*UDMP(I,J)*CONF
C
C STORE MEMBER ENERGY AND MEMBER END STATE DATA IF IFLAG=3
C
      IF(IFLAG.NE.3) GO TO 170
      URM(M)=UR
      UDM(M)=UD
     IF(ISTAT.NE.3) GO TO 170
     KR=KRAIN+21
     KS=KRESS+168
     DO 160 I=1,2
     XLOC=0.E0
     IF(I.EQ.2) XLOC=XLEN
     DO 150 K=1,11
     CALL STRN(M,XLOC,YFIBR(K,M),STRAIN)
     IF(ABS(STRAIN).LT.TINY) STRAIN=TINY
     S(9)=DATA(KR+K)
     LS=KS+8*(K-1)
     DO 135 L=1,8
135  S(L)=DATA(LS+L)
     CALL REIN (IMAT,STRAIN,URE,UDE)
     DATA(KR+K)=S(9)
     DO 140 L=1,8
140  DATA(LS+L)=S(L)
```

```
150    CONTINUE                                                              WIDE1190
       KR=KR+11                                                              WIDE1200
160    KS=KS+88                                                              WIDE1210
170    RETURN                                                                WIDE1220
C                                                                            WIDE1230
C *****  GLOSSARY  *****                                                     WIDE1240
C                                                                            WIDE1250
C   CON = TEMPORARY CONSTANT.                                                WIDE1260
C  IMAT = MATERIAL NUMBER OF WIDE FLANGE.                                    WIDE1270
C    KR = STRAIN INDEX FOR EACH FIBER.                                       WIDE1280
C    KS = STRESS INDEX FOR EACH FIBER.                                       WIDE1290
C     R = STRAIN AT MESH POINTS.                                             WIDE1300
C    UD = DISSIPATIVE ENERGY DENSITY FOR MEMBER.                             WIDE1310
C    UR = RECOVERABLE ENERGY DENSITY FOR MEMBER.                             WIDE1320
C  XLEN = LENGTH OF MEMBER.                                                  WIDE1330
C                                                                            WIDE1340
       END                                                                   WIDE1350
```

```
CZOCK    0  10
      BLOCK DATA
C
C     THIS SUBPROGRAM INITIALIZES VARIABLES STORED IN LABELED COMMON.
C
      COMMON/FIBER/DENS(9),EC(9),EPSU(9),ET(9),FCFY(9),G(9),PR(9),S(9),
     1    SLOPE(8,9),ST(17,6),STN(8,9),STS(8,9),UNLK(9),ICODE(9),NAME(9)
      COMMON/LEADBK/AVDM,AVGL,CA,CB,CC,CD,CE,DHEAD(20),DT,EPS,HEAD(20),
     1    PI,RERF,RERH,RERZ,SERR,TBEGIN,THALT,TIME,TINK,TINY,TPROB
      INTEGER HEAD,DHEAD
      COMMON/SAVEBK/SAVACC(3,50),SAVAXL(2,45),SAVCRV(2,45),SAVMOM(2,45)
     1    ,SAVSHR(2,45),SAVSRP(3,20),SAVSRQ(3,20),SAVXDJ(3,50),
     2    SAVVEL(3,50),SVSTRN(12,45),SVSTRS(12,45)
C
      DATA PI/3.14159265358 98E0/,TPROB/20.E0/
      DATA ST/33.E3,0.E0,33.E3,33.E3,49.E3,56.E3,58.E3,56.E3,2.57E4,
     1      0.E0,.114E-2, .14E-1,.59E-1,.104E0,.150E0,.210E0,8.88E-4,
     2        36.E3,0.E0,36.E3,36.E3,52.E3,58.E3,60.E3,59.E3,2.91E4,
     3     0.E0,.125E-2,.14E-1,.59E-1,.104E0,.150E0,.200E0,1.01E-3,
     4        40.E3,0.E0,40.E3,40.E3,66.E3,77.E3,80.E3,76.E3,1.49E4,
     5    0.E0,.138E-2,.230E-1,.62E-1,.101E0,.140E0,.200E0,5.14E-4,
     6        50.E3,0.E0,50.E3,50.E3,72.E3,89.E3,92.E3,90.E3,3.58E4,
     7    0.E0,.173E-2,.13E-1,.48E-1,.84E-1,.12E0,.154E0,1.237E-3,
     8        60.E3,0.E0,60.E3,60.E3,90.E3,103.E3,106.E3,100.E3,5.42E4,
     9    0.E0,.208E-2,.60E-2,.33E-1,.60E-1,.87E-1,.136E0,1.879E-3,
     $        75.E3,0.E0,75.E3,75.E3,110.E3,124.E3,130.E3,124.E3,7.5E4,
     $    0.E0,.260E-2,.27E-2,.26E-1,.50E-1,.73E-1,.115E0,2.60E-3/
      DATA SAVACC,SAVAXL,SAVCRV,SAVMOM,SAVSHR,SAVSRP,SAVSRQ,
     1 SAVXDJ,SAVVEL,SVSTRN,SVSTRS/2010*0.E0/
      END
```

DISTRIBUTION LIST

Dir.
DNA
ATTN: STSI
2 cy ATTN: SPSS
3 cy ATTN: STTL

DDR&E
ATTN: Asst. Dir., Strat. Wpns.

Dir.
DIA
ATTN: DIR-4C

Dir.
OSD, ARPA
ATTN: NMR

Cdr.
FC DNA
ATTN: FCPR

Ch.
LVLO .
ATTN: FCTCL

JSTPS
ATTN: JLTW

CO
USACDC

CO
BRL
ATTN: AMXBR-TB

Ch. of Eng.
ATTN: DAEN-RDM

Dir.
USA Eng., WWES
3 cy

Dir.
NRL
ATTN: 2027

Cdr.
NSWC
ATTN: 730

NCEL
3 cy

HQ USAF
AFTAC
ATTN: TAP

HQ USAF
ATTN: SA
ATTN: SAMI
ATTN: PREE
ATTN: PREPB
ATTN: RDPQ, 1C370
ATTN: RDQSM, 1D425
ATTN: RDQS

AFCEC
ATTN: PREC

Dir., Nuc. Surety
ATTN: SN

AFSC
ATTN: DOB

TAC
ATTN: DEE
ATTN: LGMD

CINCSAC
ATTN: DEE
ATTN: DOXS
ATTN: XPFC

AFLC
ATTN: Mr. Edward G. Fink
20 cy ATTN: DEMG

ADC
ATTN: XPQY

AU
ATTN: ED, Dir., Civ. Eng.

AFIT
ATTN: Tech. Lib., Bldg. 640, Area B
ATTN: CES

USAFA
ATTN: DFSLB
ATTN: FJSRL, CC

AFML
ATTN: Tech. Lib.

AFFDL
ATTN: DOO, Lib.

SAMSO
10 cy ATTN: DEE

DISTRIBUTION LIST (Continued).

RADC
    ATTN: Doc. Lib.

AFWL
    ATTN: DE
    ATTN: DES, 1Lt J. C. Bradshaw, III
20 cy ATTN: DES

AFOSR

Sandia Lab.
    ATTN: Org. 3141

Dir. Ofc., LLL
    ATTN: Tech. Info. Dept.

LLL
    ATTN: Lib.

Dir., LASL
    ATTN: Rpt. Lib.

Engineering Library
Air Force Institute of Technology

Engineering Library
University of California at Berkeley

Engineering Library
University of California at Los Angeles

Engineering Library
Colorado State University

Engineering Library
University of Colorado

Engineering Library
University of Illinois

Engineering Library
Massachusetts Institute of Technology

Engineering Library
University of Michigan

Engineering Library
Ohio State University

Engineering Library
Oregon State University

Engineering Library
Princeton University

Engineering Library
Stanford University

Engineering Library
Utah State University

Engineering Library
University of Utah

Department of Civil Engineering
Virginia Polytechnic Institute & State University
    ATTN: A. E. Somers
    ATTN: R. G. Galloway
50 cy ATTN: S. M. Holzer

Official Record Copy/1Lt J. C. Bradshaw, III, AFWL/DES